Водные робототехнические системы

2022/23 учебный год

Заключительный этап

Предметный тур

Информатика. 8–11 класс

Задача VI.1.1.1. Поиск трапеции (100 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 512 Мбайт.

Условие

Ваня купил себе VR-гарнитуру и решил поиграть. Для начала ему необходимо разметить VR-зону в комнате. Для этого Ваня хочет использовать изоленту. У него уже есть N отрезанных кусков длиной a_i. VR-зона должна иметь форму *прямоугольной трапеции*. Каждая сторона трапеции должна быть образована ровно одним куском изоленты.

Формат входных данных

Первая строка содержит единственное число N. Следующие N строк содержат целые числа — длины отрезков a_i.

Формат выходных данных

Выведите 4 индекса отрезков в порядке возрастания или -1, если невозможно получить прямоугольную трапецию. Индексация начинается с нуля.

Если существует несколько ответов, выведите трапецию с максимальной площадью, а среди таких — с минимальным первым индексом.

Ограничения

$1 \leqslant N \leqslant 40$, $1 \leqslant a_i \leqslant 100$.

Примеры

Пример №1

Стандартный ввод
5 12 10 3 7 23

Стандартный вывод
-1

Пример №2

Стандартный ввод
6 14 12 16 21 15 25

Стандартный вывод
1 2 4 5

Решение

Пусть дана прямоугольная трапеция с основаниями a и b и боковыми сторонами c и d, причём $a \leqslant b$ и $c \leqslant d$. Тогда должно выполняться условие $(b - a)^2 + c^2 = d^2$. Поскольку N невелико, можно перебрать все упорядоченные комбинации данных отрезков и для каждой проверить выполнение условия.

Если условие выполнено для нескольких наборов, требуется определить трапецию с минимальной площадью (площадь трапеции равна $1/2(a + b)c$, поскольку более короткая сторона в данном случае равна высоте), а среди таких — с минимальным индексом стороны.

Алгоритмическая сложность составляет $O(N^4)$.

Данную задачу можно решить и более эффективно, однако при указанных ограничениях этого не требуется.

Пример программы-решения

Ниже представлено решение на языке C++.

```
#include <iostream>
#include <cmath>
#include <set>
#include <map>
```

```cpp
#include <unordered_map>
#include <unordered_set>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <queue>
#include <string>

#define MOD 1000000007
#define INF 1000000008
#define all(arr) arr.begin(), arr.end()
#define rall(arr) arr.rbegin(), arr.rend()

#define ll long long
#define ull unsigned long long
#define double long double

#define fast ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0)
#define files(input, output) freopen(input, "r", stdin); freopen(output, "w",
↪  stdout)

using namespace std;

template<class T>
istream &operator>>(istream &in, vector<T> &arr) {
    for (T &i: arr)
        in >> i;
    return in;
}
template<class T>
ostream &operator<<(ostream &out, const vector<T> &arr) {
    for (const T &i: arr)
        out << i << ' ';
    return out;
}

bool check_trap(const vector<ll>& trap){
    ll tmp = trap[0] - trap[1];
    return tmp * tmp == trap[3] * trap[3] - trap[2] * trap[2];
}

ll area(const vector<ll> & trap){
    return (trap[0] + trap[1]) * trap[2];
}

void solve() {
    int n;
    cin >> n;
    vector<ll> segments(n);
    cin >> segments;
    ll s = 0;
    vector<ll> res(4, INF), ntrap(4), nres(4);
    for (int i = 0; i < n; ++i){
        for (int j = 0; j < n; ++j){
            if (i == j) continue;
            for (int k = 0; k < n; ++k){
                if (i == k || j == k) continue;
                for (int m = 0 ; m < n; ++m){
                    if (m == i || m == j || m == k) continue;
                    ntrap[0] = segments[i];
```

```
                    ntrap[1] = segments[j];
                    ntrap[2] = segments[k];
                    ntrap[3] = segments[m];
                    if (check_trap(ntrap)){
                        ll ns = area(ntrap);
                        nres[0] = i;
                        nres[1] = j;
                        nres[2] = k;
                        nres[3] = m;
                        sort(nres.begin(), nres.end());
                        if (ns > s) {
                            s = ns;
                            res = nres;
                        }
                        else if (ns == s){
                            bool change = false;
                            for (int l = 0; l < 4; ++l){
                                if (res[l] == nres[l]) continue;
                                if (res[l] < nres[l]){ change = false; break;}
                                change = true; break;
                            }
                            if (change){
                                res = nres;
                            }
                        }
                    }
                }
            }
        }
    }
    if (s == 0){
        cout << -1;
    }
    else{
        cout << res[0] << ' ' << res[1] << ' ' << res[2] << ' ' << res[3];
    }
}

signed main() {
    fast;
    cout.precision(20);
    int t = 1;
    while (t--)
        solve();
}
```

Задача VI.1.1.2. Экзамен (100 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 256 Мбайт.

Условие

Артему предстоит сложный экзамен по алгебре, геометрии и физике. Преподаватель на лекции сообщил, что студентам на выбор будет предложено n билетов и скинул всем их на электронную почту. Артем оценил каждый билет по трем параметрам: сложность его по алгебре, геометрии и физике. Сам Артем оценил свои силы следующим образом: если сложность билета по алгебре, геометрии и физике меньше чем a, b и c соответственно, он способен к нему подготовится и сдать.

Так как Артем был прилежным студентом, преподаватель готов принять у него экзамен, пропустив один из предметов по его выбору: алгебру, геометрию или физику. Студент не против воспользоваться такой возможностью, поэтому просит у вас узнать, какой предмет ему стоит пропустить, чтобы можно было решить как можно больше билетов.

Формат входных данных

В первой строке записано три целых числа a, b и c — предельные пороги сложности по алгебре, геометрии и физике соответственно.

Во второй строке целое число n — количество билетов на экзамене.

Во следующих n строках записано по три целых числа a_i, g_i, p_i — сложность билета по алгебре,геометрии и физике соответственно.

Формат выходных данных

Выведите `Algebra`, `Geometry` или `Physics` в соответствии с тем, какой экзамен выгоднее пропустить. Если существует несколько вариантов ответа, выведите любой.

Ограничения

$1 \leqslant n \leqslant 10^5$.

$1 \leqslant a, b, c, a_i, b_i, c_i \leqslant 10^9$.

Примеры

Пример №1

Стандартный ввод
3 3 3 3 2 4 5 2 2 5 1 1 4

Стандартный вывод
Physics

Пример №2

Стандартный ввод
5 2 4 5 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5

Стандартный вывод
Geometry

Решение

Данная задача является тривиальной и имеет линейное решение $O(n)$.

Посчитаем для каждого предмета, сколько билетов можно сдать успешно, пропустив выбранный предмет. Подсчет будем осуществлять следующим образом: если пропускаем a (алгебра), считаем количество билетов, для которых $b > b_i$ (геометрия) и $c > c_i$ (физика). Для остальных предметов делается также.

Далее мы должны найти и вывести тот предмет, при котором получилось максимальное количество сданных билетов.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

struct exam {
    int a;
    int b;
    int c;
};

int main() {
    int a, b, c, n, resA = 0, resB = 0, resC = 0;
    cin >> a >> b >> c >> n;
    vector<exam> arr(n);
    for(int i = 0; i < n; i++)
        cin >> arr[i].a >> arr[i].b >> arr[i].c;
    for(int i = 0; i < n; i++) {
        if(b >= arr[i].b && c >= arr[i].c)  resA++;
        if(a >= arr[i].a && c >= arr[i].c)  resB++;
        if(a >= arr[i].a && b >= arr[i].b)  resC++;
    }
    int maxValue = max(resA, max(resB, resC));
    cout << "Possible answers: ";
    if(maxValue == resA)
```

```
            cout << "Algebra ";
        if(maxValue == resB)
            cout << "Geometry ";
        if(maxValue == resC)
            cout << "Physics ";
}
```

Задача VI.1.1.3. Радостные студенты (100 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 512 Мбайт.

Условие

На лекции по высшей математике в ДВФУ преподаватель собрал n студентов в ряд и задал простой вопрос: «*Кто сейчас грустит, поднимите руку*». На это предложение несколько (возможно, ноль) студентов подняли руки.

После этого он решил выбрать из этой последовательности студентов некоторый отрезок `[left, right]`, на котором он добавит грустным студентам по 5 баллов к экзамену просто так. При этом он понимает, что студенты, которые были радостные на этом отрезке, поменяют своё настроение. Если ни один студент не является грустным, преподаватель не будет выбирать никакой отрезок. Он хочет получить наибольшее количество радостных студентов на лекции, поэтому просит вас написать программу, которая рассчитает максимальное их количество после применения ранее описанной операции.

Формат входных данных

В первой строке записано целое число n — количество студентов.

Во второй строке записано n цифр 0 и 1, где 0 — грустный студент, а 1 — радостный.

Формат выходных данных

Выведите максимально возможное количество радостный студентов.

Ограничения

$1 \leqslant n \leqslant 2 \cdot 10^5$.

Критерии оценивания

Баллы начисляются за каждый тест независимо. Тесты поделены по подзадачам, описанным ниже.

Подзадача	Количество тестов	Баллы	Дополнительные ограничения	Информация о проверке
			n	
1	5 тестов	5 баллов за тест	$1 \leqslant n \leqslant 200$	полная
2	5 тестов	5 баллов за тест	$1 \leqslant n \leqslant 2000$	полная
3	10 тестов	5 баллов за тест	$1 \leqslant n \leqslant 2 \cdot 10^5$	полная

Примеры

Пример №1

Стандартный ввод
7 1 1 1 0 0 1 0

Стандартный вывод
6

Пример №2

Стандартный ввод
5 1 1 1 1 1

Стандартный вывод
5

Решение

Идея решения данного задания — линейный поиск максимальной суммы подотрезка.

Посчитаем первоначальное количество радостных студентов. Следующим шагом (или параллельно) создадим массив, содержащий изменения настроения студентов, если они войдут в подотрезок профессора. Радостных студентов обозначим как -1 (настроение станет негативным), а грустных как 1 (настроение станет позитивным).

Следующим шагом создадим две переменные, в которых будут хранится сумма изменений студентов на некотором отрезке $[1, i]$ и сумма изменений настроений на некотором отрезке $[1, j]$, $j < i \leqslant n$. Параметр j будет на каждом шагу выбираться так, чтобы сумма отрезка $[1, j]$ являлась наименьшей на отрезке $[1, i]$.

Отрезок минимальной суммы будет убираться из отрезка общей суммы $sum[1, i] - sum[1, j]$, тем самым мы сможем получить наибольшую сумму изменений настроений студентов. Так как мы изначально сделали радостных студентов -1, а грустных 1, процесс вычисления будет стараться избегать случаем, когда изменится настроение у позитивных студентов, но при этом будет искать наибольшее количество грустных студентов и компенсировать настроение новоиспеченных грустных студентов.

Среди всех подотрезков выберем тот, у которого наибольшее изменение настроений, добавим к уже ранее высчитанным радостным студентам, и выведем полученную сумму. Данный результат будет верным, ведь если изменится настроение у радостного студента, наш подотрезок будет это учитывать. Например, если на подотрезке 3 грустных студента и 1 радостный, то сумма настроений на подотрезке будет 2 (двое станут новыми радостными, а оставшиеся два студента, радостный и грустный, компенсируют свои настроения).

Пример программы-решения

Ниже представлено решение на языке C++.

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> arr (n);
    int res = 0;
    for(auto &i : arr) {
        cin >> i;
        if(i == 0) {
            i = 1;
        } else {
            i = -1;
            res++;
        }
    }
    int ans = 0;
    int sum = 0;
    int minSum = 0;
    for(int i = 0; i < n; i++) {
        sum += arr[i];
        ans = max(ans, sum - minSum);
        minSum = min(minSum, sum);
    }
    cout << res + ans;
}
```

Задача VI.1.1.4. Подготовка к ЕГЭ (100 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 512 Мбайт.

Условие

Мальчик Миша готовится к экзаменам. На это у него осталось N дней. В i-ый день у Миши вдохновение решить a_i задач. Но он не сверхчеловек, поэтому ему необходимо спать. В i-ый день у Миши есть выбор:

- Поспать, не решив ни одной задачки.

- Выпить чай с лимонником и не спать, решив все задачи.
- Не спать, решив все задачи.

Не спать он может только в том случае, если у него достаточно сил, то есть если в предыдущий день он поспал.

Также у него есть замечательный напиток — чай с лимонником, который даст ему сил не спать. Но при этом Миша знает, что избыток чая вреден для здоровья, поэтому он не станет его пить, если делал это в предыдущий день.

Формат входных данных

В первой строке записано целое число N.

Во второй строке находится N целых чисел a_i.

Формат выходных данных

Выведите одно целое число: максимальное количество задач, которые может решить Миша.

Ограничения

$1 \leqslant N \leqslant 2 \cdot 10^5$.

$1 \leqslant a_i \leqslant 10^6$.

Критерии оценивания

Баллы начисляются за каждый тест независимо. Тесты поделены по подзадачам, описанным ниже.

Подзадача	**Количество тестов**	**Баллы**	**Дополнительные ограничения**	**Информация о проверке**
			N	
1	5 тестов	4 балла за тест	$1 \leqslant N \leqslant 20$	полная
2	5 тестов	4 балла за тест	$1 \leqslant N \leqslant 5 \cdot 10^4$	полная
3	10 тестов	4 балла за тест	$1 \leqslant N \leqslant 2 \cdot 10^5$	полная

Примеры

Пример №1

Стандартный ввод
3 1 2 3

Стандартный вывод
5

Пример №2

Стандартный ввод
5 1 1 1 1 1

Стандартный вывод
4

Решение

Решим задачу при помощи динамического программирования. Создадим 3 массива длины N, где в первом на i-ой позиции будет храниться максимальное количество задач, решенное к i-ому дню включительно, если в этот день Миша решит поспать. Во втором — если решит попить чай. А в третьем — если решит не спать.

База динамики: в первом массиве на первой позиции будет стоять 0, так как Миша будет спать и не решит задач. Во втором и третьем массивах на первой позиции будет стоять a_1.

Далее для всех i от 2 до N на i-ой позиции:

- первого массива — максимум из значений всех массивов на позиции $i - 1$;
- второго массива — сумма a_i и максимума из значений первого и третьего массивов на позиции $i - 1$;
- третьего массива — сумма значения первого массива на $i - 1$ позиции и a_i.

Ответом будет максимум из элементов на N-ой позиции всех трёх массивов.

Пример программы-решения

Ниже представлено решение на языке C++.

```
#include "iostream"
#include "vector"

using namespace std;

int main() {
    int N;
    cin >> N;
    vector<long long> a(N);
    for (int i = 0; i < N; ++i) {
        cin >> a[i];
    }
    vector<vector<long long>> dp(N, vector<long long>(3, 0)); //спать, чай, не спать
    dp[0][0] = 0;
    dp[0][1] = a[0];
    dp[0][2] = a[0];
    for (int i = 1; i < N; ++i) {
        dp[i][0] = max(max(dp[i - 1][0], dp[i - 1][1]), dp[i - 1][2]);
        dp[i][1] = max(dp[i - 1][0], dp[i - 1][2]) + a[i];
        dp[i][2] = dp[i - 1][0] + a[i];
    }
    cout << max(dp[N - 1][0], max(dp[N - 1][1], dp[N - 1][2]));
}
```

Задача VI.1.1.5. Путешествие (100 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 512 Мбайт.

Условие

Утенок Даки только выпустился из университета и устроился на работу разработчиком игр. Ему поручили создание новой игры SpaceWar. Игра заключается в сражениях с космическим флотом противника путём отправки флотов кораблей между планетами.

Даки закончил разработку и решил проверить качество игры самостоятельно. В ходе проверки утенок понял, что в игре играет роль не только мощь флота, но и распределение его между планетами, чтобы вражеский флот не смог захватывать территории. Даки решил опробовать новую тактику и, несмотря на опасность потери планет, не разделять свой флот. Так он сможет с легкостью захватывать планеты. Но такая тактика имеет большой недостаток: во время нападения на новые планеты собственные территории беззащитны. Поэтому утенок хочет рассчитывать время, за которое его флот сможет добраться до планеты.

В этой игре между двумя планетами иногда появляются «кротовые норы» (англ. wormholes), пройдя через которые, флот может переместиться вперед во времени и оказаться у следующей планеты. В то же время может также существовать «обычный» путь между планетами, который флот может преодолеть за некоторое время. Существование кротовых нор и обычного пути не связаны между собой. Кротовые норы появляются не сразу и, если флот окажется у планеты, нора около которой еще не образовалась, он не сможет ею воспользоваться. Воспользоваться норой в обратном направлении невозможно. Даки хочет определить, в какой самый ранний момент времени его флот, находящийся у планеты A, может оказаться у планеты B.

Формат входных данных

Первая строка содержит 3 целых числа: N, A, B — количество планет, планету с флотом и планету-цель, соответственно.

Далее следует строка с 2 целыми числами: M — количество кротовых нор и K — количество обычных путей.

Последующие M строк содержат 4 целых числа: A_i, B_i — две планеты, между которыми появляется нора, t_i — время её появления и dt_i — насколько изменится время при перемещении по ней из A_i в B_i.

Далее идут K строк в таком формате: A_j, B_j, t_j — две планеты и время пути между ними.

Считаем, что изначально время равно 0. Все планеты нумеруются от 1 до N включительно. Обычные пути существуют в любое время.

Формат выходных данных

Выведите единственное число — самое раннее время в которое флот сможет оказаться у планеты B. Гарантируется, что путь существует.

Ограничения

$1 \leqslant A, B, A_i, B_i, A_j, B_j \leqslant N \leqslant 10^4$

$N \leqslant M + K \leqslant 10^5$

$0 \leqslant t_i, dt_i, t_j \leqslant 10^9$

Примеры

Пример №1

Стандартный ввод

```
6 3 5
3 6
6 3 0 0
1 3 2 3
2 1 0 1
3 5 3
1 6 2
5 1 4
3 6 0
5 2 1
2 4 2
```

Стандартный вывод

```
3
```

Пример №2

Стандартный ввод

```
5 3 2
0 8
3 2 4
1 4 1
5 2 2
5 3 5
1 5 3
2 4 1
4 1 3
4 3 2
```

Стандартный вывод

```
4
```

Решение

Для решения задачи можно воспользоваться модифицированным алгоритмом Дейкстры. Длина пути в данном случае эквивалентна самом раннему времени прибытия.

Напомним, что в этом алгоритме вершины почерёдно помечаются как обработанные, что означает что текущее найденное расстояние до них является кратчайшим.

На этапе релаксации пути помимо рёбер, исходящих из только что помеченной вершины, следует также рассмотреть новые рёбра, образовавшиеся при появлении «кротовых нор». При этом в качество новой длины пути следует брать сумму времени перемещения по норе и минимума из времени прибытия в помеченной вершину и времени появления норы.

Для повышения эффективности выбора помечаемой вершины можно использовать бинарную кучу.

Пример программы-решения

Ниже представлено решение на языке C++.

```cpp
#include <iostream>
#include <cmath>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <queue>
#include <string>

#define MOD 1000000007
#define INF 1000000008
#define all(arr) arr.begin(), arr.end()
#define rall(arr) arr.rbegin(), arr.rend()

#define ll long long
#define ull unsigned long long
#define double long double

#define fast ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0)
#define files(input, output) freopen(input, "r", stdin); freopen(output, "w",
↪ stdout)

using namespace std;

template<class T>
istream &operator>>(istream &in, vector<T> &arr) {
    for (T &i: arr)
        in >> i;
    return in;
}

template<class T1, class T2>
istream &operator>>(istream &in, pair<T1, T2> &p) {
    in >> p.first >> p.second;
    return in;
}
struct hole{
    int from;
```

```
    int to;
    int t;
    int dt;
};
void solve() {
    int n, a, b;
    cin >> n >> a >> b;
    --a; --b;
    vector<vector<pair<int, int>>> ways(n, vector<pair<int, int>>());
    vector<vector<hole>> holes(n, vector<hole>());
    vector<ull> d(n, INF);
    d[a] = 0;
    vector<pair<ull, int>> heap = {{0,a}};
    make_heap(heap.begin(), heap.end(), std::greater<>{});
    int m, k;
    cin >> m >> k;
    for (int i = 0; i < m; ++i){
        hole tmp{};
        cin >> tmp.from>> tmp.to >> tmp.t >> tmp.dt;
        --tmp.from; --tmp.to;
        holes[tmp.from].push_back(tmp);
    }
    for (int j = 0; j < k; ++j){
        int a_j, b_j, tmp;
        cin >> a_j >> b_j >> tmp;
        --a_j; --b_j;
        ways[a_j].emplace_back(b_j, tmp);
        ways[b_j].emplace_back(a_j, tmp);
    }
    while (d[b] == INF) {
        if (heap.empty())
            break;
        pop_heap(heap.begin(), heap.end(), std::greater<>{});
        pair<ull, int> v = heap.back();
        heap.pop_back();
        if (d[v.second] != INF && v.second != a) continue;
        d[v.second] = v.first;
        for (const auto &w: ways[v.second]) {
            heap.emplace_back(d[v.second] + w.second, w.first);
            push_heap(heap.begin(), heap.end());
        }
        for (const auto &h: holes[v.second]) {
            ll tmp;
            if (h.t <= d[v.second]) {
                tmp = d[v.second] + h.dt;
            } else {
                tmp = (ll) h.t + h.dt;
            }
            heap.emplace_back(tmp, h.to);
            push_heap(heap.begin(), heap.end());
        }
    }
    cout << d[b];
}

signed main() {
    fast;
    cout.precision(20);
    int t = 1;
    while (t--)
        solve();
}
```

Физика. 8–9 классы

Задача VI.1.2.1. Глубоководный аппарат (15 баллов)

Темы: *кинематика равномерного движения.*

Условие

Средняя скорость глубоководного аппарата при спуске на глубину H и подъёме на поверхность $v_{\text{ср}} = 2,0$ км/ч. Скорость спуска аппарата v_1 в 1,3 раза больше, чем скорость его подъёма v_2, которая в свою очередь составляет 60% от скорости v_0 движения аппарата на фиксированной глубине. Определите:

1. скорость v_0 глубоководного аппарата на фиксированной глубине;
2. время $t_{\text{п}}$, за которое аппарат поднимается с глубины $h_{\text{п}} = 1$ км;
3. глубину $h_{\text{с}}$, на которую аппарат спускается за время $t_{\text{с}} = 40$ мин.

Решение

Так как скорость спуска аппарата v_1 в 1,3 раза больше, чем скорость его подъёма v_2, то $v_1 = 1,3v_2$.	1 балл
Время спуска аппарата на глубину H $t_1 = H/v_1$.	1 балл
Время подъёма аппарата с глубины H $t_2 = H/v_2$.	1 балл
Общее время спуска и подъёма аппарата на глубину H $t = t_1 + t_2 = H/v_1 + H/v_2 = H(v_1 + v_2)/v_1v_2 = 1,77H/v_2$.	2 балла
Средняя скорость аппарата при спуске и подъёме $v_{\text{ср}} = 2H/t$.	2 балла
Из уравнения $v_{\text{ср}} = 2H/t = 1,13v_2$ определим скорость подъёма аппарата $v_2 = v_{\text{ср}}/1,13 = 1,77$ км/ч.	1 балл
Аппарат поднимается с глубины $h_{\text{п}}$ за время $t_{\text{п}} = h_{\text{п}}/v_2 = 0,56$ ч $= 34$ мин.	2 балла
Скорость аппарата на фиксированной глубине $v_0 = v_2/0,6 = 2,95$ км/ч.	2 балла
Скорость спуска аппарата $v_1 = 1,3v_2 = 2,3$ км/ч.	1 балл
За время $t_{\text{с}}$ аппарат спустится на глубину $h_{\text{с}} = v_1 tc = 1,53$ км.	2 балла

Ответ: $v_0 = 2,95$ км/ч; $t_{\text{п}} = 34$ мин; $h_{\text{с}} = 1,53$ км.

Задача VI.1.2.2. Электрическая цепь (15 баллов)

Темы: *закон Ома для участка цепи.*

Условие

Девять резисторов с сопротивлениями $R_1 = 1$ Ом, $R_2 = 2$ Ом, $R_3 = 3$ Ом, ..., $R_8 = 8$ Ом, $R_9 = 9$ Ом соединили последовательно и подключили к источнику тока напряжением $U_0 = 13{,}5$ В. Определите общее сопротивление R_0 цепи и силу тока I_0 в цепи. В соединения цепи между резисторами R_2, R_3 и резисторами R_6, R_7 подключили идеальный амперметр, а между резисторами R_4, R_5 и резисторами R_8, R_9 подключили идеальный вольтметр. Определите показания приборов I_A и U_V.

Решение

Общее сопротивление цепи равно $R_0 = R_1 + R_2 + \dots + R_8 + R_9 = 45$ Ом.	2 балла
Общий ток в цепи $I_0 = U_0/R_0 = 0{,}3$ А.	3 балла
При подключении идеального амперметра в соединения цепи между резисторами R_2, R_3 и резисторами R_6, R_7 ток через резисторы R_3, R_4, R_5 и R_6 течь не будет, поэтому показания амперметра будет равно $I_A = U_0/(R_1 + R_2 + R_7 + R_8 + R_9) = 0{,}5$ А.	5 баллов
Так как через резисторы R_5 и R_6 ток не течёт, то вольтметр, подключённый между резисторами R_4, R_5 и резисторами R_8, R_9, покажет суммарное напряжение на резисторах R_7 и R_8: $U_V = I_A(R_7 + R_8) = 7{,}5$ В.	5 баллов

Ответ: $R_0 = 45$ Ом; $I_0 = 0{,}3$ А; $I_A = 0{,}5$ А; $U_V = 7{,}5$ В.

Задача VI.1.2.3. Неравноплечные весы (20 баллов)

Темы: статика, правило моментов.

Условие

Однородный железный стержень длиной $l = 1{,}6$ м и площадью сечения $S = 500$ мм^2 подвесили так, что в точке подвеса длина стержня делится в пропорции $2 : 3$. Определите массу m_1 груза, который необходимо подвесить к одному из концов стержня, чтобы стержень находился в состоянии равновесия (принял горизонтальное положение). Далее груз массой m_1 перевешивают на противоположный конец стержня. Определите массу m_2 груза, который необходимо подвесить к свободному концу стержня, чтобы стержень опять принял состояние равновесия. Плотность железа $\rho = 7{,}8$ г/см^3.

Решение

Масса железного стержня $M = \rho S l = 6{,}24$ кг.	2 балла

Центр масс стержня будет находиться от точки подвеса на расстоянии $l_0 = 1/2l - 2/5l = 1/10l = 16$ см.	2 балла
Груз массы m_1 необходимо подвесить к ближайшему к точке подвеса концу стержня.	2 балла
Из условия равновесия стержня $m_1 g \cdot 2/5l = Mg \cdot l_0$	4 балла
Определим $m_1 = M/4 = 1,56$ кг.	3 балла
Из нового условия равновесия стержня $m_2 g \cdot 2/5l = Mg \cdot l0 + m_1 g \cdot 3/5l$	4 балла
Определим $m_2 = (M + 6m_1)/4 = 5M/8 = 3,9$ кг.	3 балла

Ответ: $m_1 = 1,56$ кг; $m_2 = 3,9$ кг.

Задача VI.1.2.4. Поплавок (25 баллов)

Темы: гидростатика, сила Архимеда.

Условие

Поплавок объёмом $V = 5000$ см^3, имеющий воздушную полость, плавает в воде, погрузившись на 45% своего объёма. Если полость заполнить глицерином, то поплавок будет плавать в воде, погрузившись в неё на 75% своего объёма. Определите объём полости $V_п$, плотность материала ρ, из которого сделан поплавок, и массу m поплавка, если плотность воды $\rho_в = 1000$ кг/м^3, плотность глицерина $\rho_г = 1200$ кг/м^3. Какую часть полости необходимо заполнить водой, чтобы поплавок плавал в керосине, полностью в него погрузившись? Плотность керосина $\rho_к = 700$ кг/м^3.

Решение

Условие плавания поплавка в воде: $\rho(V - V_п)g = 0,45\rho_в V g$.	2 балла
Условие плавания поплавка с заполненной глицерином полостью в воде: $\rho(V - V_п)g + \rho_г V_п g = 0,75\rho_в V g$.	3 балла
Решая систему записанных уравнений, определим объём полости $V_п = 0,3\rho_в V/\rho_г = 1250$ см^3	5 баллов
и плотность поплавка $\rho = 0,45\rho_в\rho_г/(\rho_г - 0,3\rho_в) = 600$ кг/м^3.	5 баллов
Масса поплавка $m = \rho(V - V_п) = 2,25$ кг.	2 балла
Из условия плавания поплавка с частично заполненной водой полостью в керосине $mg + x\rho_в V_п g = \rho_к V g$.	3 балла

Определим часть полости, заполненной водой: $x = (\rho_к V - m)/\rho_в V_п = 1$,	3 балла
то есть полость необходимо заполнить водой полностью.	2 балла

Ответ: Vп $= 1250$ см^3; $\rho = 600$ кг/м^3; $m = 2,25$ кг; $x = 1$.

Задача VI.1.2.5. Вода и лёд (25 баллов)

Темы: тепловой баланс.

Условие

В калориметре в тепловом равновесии находится вода $m_в = 210$ г и лёд $m_л = 10$ г. В калориметр помещают алюминиевый шарик массой $m_а = 185$ г при температуре $t_а = -10$ °C. Определите температуру $t_{р1}$, массу воды $m_{в1}$ и льда $m_{л1}$ в равновесном состоянии. Далее в калориметр помещают железный шарик массой $m_ж = 500$ г при температуре $t_ж = 80$ °C. Определите температуру $t_{р2}$, массу воды $m_{в2}$ и льда $m_{л2}$ в конечном состоянии. Удельная теплоёмкость воды $c_в = 4200$ Дж/(кг·°C), льда — $c_л = 2100$ Дж/(кг·°C), алюминия — $c_а = 920$ Дж/(кг·°C), железа — $c_ж = 460$ Дж/(кг·°C), удельная теплота плавления льда $\lambda = 340$ кДж/кг. Теплоёмкостью калориметра и тепловыми потерями пренебречь.

Решение

Вода и лёд находятся в тепловом равновесии при температуре $t_0 = 0$ °C.	2 балла
Тепло, необходимое для нагревания алюминиевого шарика до температуры t_0: $Q = m_а c_а (t_0 - t_а) = 1,7$ кДж.	2 балла
Это тепло будет получено при замерзании воды, массой $m = Q/\lambda = 5$ г.	2 балла
Масса воды в калориметре: $m_{в1} = m_в - m = 205$ г,	2 балла
масса льда в калориметре: $m_{л1} = m_л + m = 15$ г.	2 балла
Так как в калориметре осталась смесь воды и льда, то $t_{р1} = t_0 = 0$ °C	2 балла
Тепло, выделившееся при охлаждении железного шарика до температуры t_0: $Q_1 = m_ж c_ж (t_ж - t_0) = 18,4$ кДж.	2 балла
Тепло, необходимое для плавления всего льда: $Q_2 = m_{л1}\lambda = 5,1$ кДж.	2 балла
Избыток тепла $\Delta Q = Q_1 - Q_2 = 13,3$ кДж пойдёт на нагревание всей воды (которая была и которая образовалась изо льда), алюминиевого и железного шариков до температуры теплового равновесия $t_{р2}$: $\Delta Q = ((m_{в1} + m_{л1})c_в + m_а c_а + m_ж c_ж)(t_{р2} - t_0)$.	2 балла

Таким образом, $$t_{\text{р}2} = \frac{\Delta Q}{(m_{\text{в}1} + m_{\text{л}1})c_{\text{в}} + m_{\text{а}}c_{\text{а}} + m_{\text{ж}}c_{\text{ж}}} = 10\ ^\circ\text{C}$$.	3 балла
Конечная масса воды в калориметре: $m_{\text{в}2} = m_{\text{в}1} + m_{\text{л}1} = 220$ г,	2 балла
конечная масса льда в калориметре: $m_{\text{л}2} = 0$ г.	2 балла

Ответ: $t_{\text{р}1} = 0\ ^\circ\text{C}$, $m_{\text{в}1} = 205$ г, $m_{\text{л}1} = 15$ г;
$t_{\text{р}2} = 10\ ^\circ\text{C}$, $m_{\text{в}2} = 220$ г, $m_{\text{л}2} = 0$ г

Физика. 10–11 классы

Задача VI.1.3.1. Батискаф (15 баллов)

Темы: теплопередача.

Условие

Для поддержания постоянной температуры в батискафе используется нагревательный элемент, имеющий три спирали одинаковой мощности. При включении одной спирали температура воздуха в батискафе равна $t_1 = 14\ ^\circ\text{C}$. При включении всех трёх спиралей нагревателя температура в батискафе увеличивается до $t_2 = 26\ ^\circ\text{C}$. Найдите температуру воды t_0 за бортом батискафа, считая ее постоянной. Мощность теплопередачи пропорциональна разности температур воздуха в батискафе и окружающей его воды: $P = \alpha\Delta t$. Какой будет температура в батискафе, если включить две спирали нагревательного элемента?

Решение

Обозначим: P — мощность одной спирали нагревательного элемента. Уравнение теплового баланса при включении одной спирали нагревателя: $P = \alpha(t_1 - t_0)$.	3 балла
Уравнение теплового баланса при включении трёх спиралей нагревателя: $3P = \alpha(t_2 - t_0)$.	3 балла
Из уравнения $3\alpha(t_1 - t_0) = \alpha(t_2 - t_0)$ найдём $t_0 = (3t_1 - t_2)/2 = 8\ ^\circ\text{C}$.	3 балла
Уравнение теплового баланса при включении двух спиралей нагревателя: $2P = \alpha(t_3 - t_0)$.	3 балла
Из уравнения $\alpha(t_3 - t_0) = 2\alpha(t_1 - t_0)$ найдём $t_3 = 2t_1 - t_0 = (t_1 + t_2)/2 = 20\ ^\circ\text{C}$.	3 балла

Ответ: $t_0 = 8$ °C, $t_3 = 20$ °C.

Задача VI.1.3.2. Всплытие шаров (15 баллов)

Темы: динамика, силы Архимеда и сопротивления.

Условие

Определите, с какой скоростью v_0 будет всплывать в воде шар плотностью $\rho_0 = 400$ кг/м^3 и объёмом $V_0 = 0,25$ м^3, если сила сопротивления воды при движении шара пропорциональна его скорости $F_c = kv$, $k = 750$ кг/с. С какими скоростями v_1 и v_2 будут всплывать в воде два шара, связанные длинным тонким невесомым нерастяжимым тросом при установившемся движении, если второй шар имеет нулевую плавучесть (находится в безразличном равновесии в воде, погружаясь в неё полностью)? Плотность воды $\rho_в = 1000$ кг/м^3.

Решение

При движении одного шара на него действуют следующие силы: $m_0 g = \rho_0 V_0 g$ — сила тяжести; $F_A = \rho_в V_0 g$ — сила Архимеда и сила сопротивления $F_c = kv$. В установившемся режиме шар будет всплывать с постоянной скоростью v_0, при этом все силы, действующие на шар, должны быть скомпенсированы. Из уравнения $F_A - m_0 g - F_c = 0$	2 балла
определим скорость движения шара $v0 = (\rho_в - \rho_0)Vg/k = 2$ м/с.	3 балла
При всплытии двух шаров, связанных тросом, в установившемся движении в воде шары расположатся на одной вертикали, при этом легкий шар окажется выше тяжелого, а трос будет натянут. Вследствие нерастяжимости троса шары будут двигаться с равными скоростями: $v_1 = v_2 = v$.	2 балла
На шары кроме силы тяжести, силы Архимеда, силы сопротивления воды будет действовать сила натяжения троса T. Запишем уравнение движения каждого шара: $F_A - m_0 g - F_c - T = 0$,	2 балла
$F_{A2} - m_2 g - F_c + T = 0$,	2 балла
где $m_2 g$ (F_{A2}) — сила тяжести (Архимеда), действующая на второй шар. Так как второй шар имеет нулевую плавучесть, $F_{A2} = m_2 g$,	1 балл
То для силы натяжения троса получим: $T = F_c = kv_1 = kv_2 = kv$.	1 балл
Таким образом, шары будут всплывать со скоростью $v = v_0/2 = 1$ м/с.	2 балла

Ответ: $v_0 = 2$ м/с; $v_1 = v_2 = 1$ м/с.

Задача VI.1.3.3. Пружинка (20 баллов)

Темы: *сила упругости, работа, энергия.*

Условие

К пружинке жёсткостью $k = 200$ Н/м подвесили груз массой $m = 500$ г. Верхний конец пружинки жёсткостью $k = 200$ Н/м закрепили, а к нижнему подвесили груз массой $m = 500$ г. Определите удлинение пружинки Δl_0. Далее груз стали медленно перемещать, прикладывая к нему силу, направленную вертикально. Определите,

1. какую работу A_1 необходимо совершить, чтобы удлинение пружины увеличить в три раза;
2. какую работу A_2 необходимо совершить, чтобы из начального положения пружину сжать на Δl_0?

Ускорение свободного падения $g = 10$ м/с^2; массой пружинки пренебречь.

Решение

Из условия равновесия груза, подвешенного к пружине $mg = k\Delta l_0$,	3 балла
получим $\Delta l_0 = mg/k = 2,5$ см.	3 балла
Работа силы при увеличении удлинения пружины идёт на изменение потенциальной энергии пружины $\Delta E_п = k(\Delta l_1^2 - \Delta l_0^2)/2$	2 балла
и потенциальной энергии груза: $\Delta E_г = -mg(\Delta l_1 - \Delta l_0)$.	2 балла
Так как $\Delta l_1 = 3\Delta l_0$, то $A_1 = \Delta E_п + \Delta E_г = 4k\Delta l_0^2 - 2mg\Delta l_0 = 2(mg)^2/k = 0,25$ Дж.	4 балла
Так как во втором случае потенциальная энергия пружины не меняется, то работа силы идёт только на изменение потенциальной энергии груза $A_2 = mg2\Delta l_0 = 2(mg)^2/k = 0,25$ Дж.	6 баллов

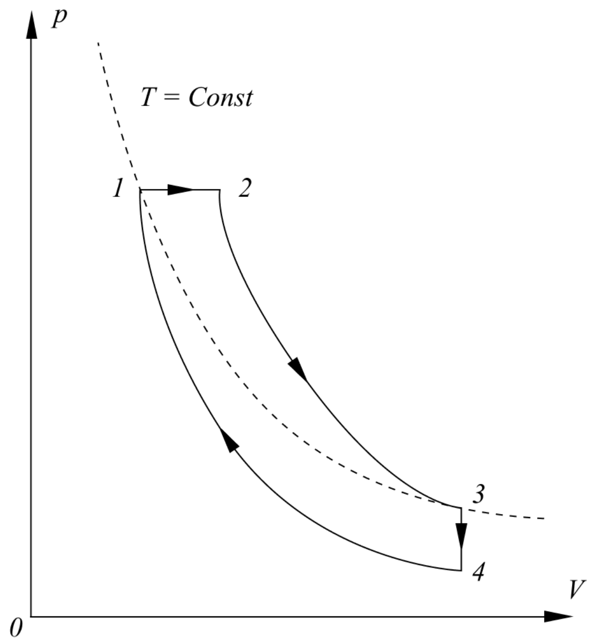
Примечание: при непосредственном расчёте работы силы необходимо учитывать, что внешняя сила увеличивается по линейному закону, поэтому $F_{ср} = F_{max}/2$;
$A = F_{ср} \cdot S = 1/2(k2\Delta l_0) \cdot 2\Delta l_0 = 2k\Delta l_0^2 = 2(mg)^2/k$.

Ответ: $\Delta l_0 = 2,5$ см; $A_1 = 0,25$ Дж; $A_2 = 0,25$ Дж.

Задача VI.1.3.4. Замкнутый цикл (25 баллов)

Темы: *термодинамика, коэффициент полезного действия.*

Одноатомный идеальный газ совершает замкнутый цикл, состоящий из изобары ($1 \to 2$), изохоры ($3 \to 4$), и двух адиабат ($2 \to 3$ и $4 \to 1$). Точки 1 и 3 находятся на одной изотерме. Найдите температуру T, соответствующую изотерме 1–3, если температуры точек 2 и 4 равны $T_2 = 490$ К и $T_4 = 350$ К, соответственно, а КПД цикла равен $\eta = 20\%$.



Решение

Запишем уравнения состояния газа для точек $1, \ldots, 4$: $p_i V_i = \nu R T_i, \;\; i = 1, 2, 3, 4.$	1 балл
Для изобары имеем $p_1 = p_2$.	1 балл
Для изохоры имеем $V_3 = V_4$.	1 балл
Так как точки 1 и 3 находятся на одной изотерме, то $T_1 = T_3 = T$.	1 балл
Газ получает тепло при изобарическом нагревании: $Q_+ = Q_{12} = A_{12} + \Delta U_{12}$.	1 балл
Работа газа на участке $1 \to 2$: $A_{12} = p_1(V_2 - V_1) = \nu R(T_2 - T_1)$.	2 балла
Изменение внутренней энергии газа на участке $1 \to 2$: $\Delta U_{12} = 3/2 \nu R(T_2 - T_1)$.	1 балл
Таким образом, $Q_+ = 5/2 \nu R(T_2 - T_1)$.	2 балла
Газ отдаёт тепло при изохорическом охлаждении: $Q_- = Q_{34} = A_{34} + \Delta U_{34}$.	1 балл

Работа газа на участке $3 \to 4$: $A_{34} = 0$, так как $V_3 = V_4$.	1 балл
Изменение внутренней энергии газа на участке $3 \to 4$: $\Delta U_{34} = 3/2\nu R(T_4 - T_3)$.	1 балл
Таким образом, $Q_- = 3/2\nu R(T_4 - T_3)$.	2 балла
Так как участки $2 \to 3$ и $4 \to 1$ адиабаты, то $Q_{23} = Q_{41} = 0$.	2 балла
КПД цикла $\eta = 1 - \lvert Q_-\rvert/Q_+$.	3 балла
Подставляя в это уравнение выражения для Q_+ и Q_-, получим $$T = \frac{3T_4 + 5(1-\eta)T_2}{8 - 5\eta} = 430\ \text{К}.$$	5 баллов

Ответ: $T = 430$ К.

Задача VI.1.3.5. Батарейка и резисторы (25 баллов)

Темы: закон Ома для полной цепи, мощность.

Условие

К батарейке с ЭДС $\mathcal{E} = 4,5$ В поочерёдно подключают резисторы с сопротивлениями $R_1 = 3,75$ Ом и $R_2 = 1,5$ Ом. При этом количество теплоты, выделяющееся в единицу времени резисторе R_2, в 1,6 раза больше, чем количество теплоты, выделяющееся в единицу времени резисторе R_1. Определите внутренне сопротивление r батарейки. При каком сопротивлении R_3 резистора на нём будет выделяться наибольшая мощность тепла? Чему равна наибольшая мощность P_{max}?

Решение

При подключении к батарейке с ЭДС $\mathcal{E}$ и внутренним сопротивлением r резистора с сопротивлением R в цепи течёт ток $I = \mathcal{E}/(R + r)$.	2 балла
Тепло, выделяющееся в единицу времени на резисторе (мощность): $Q/\tau = P = I^2R = \mathcal{E}^2R/(R + r)^2$.	2 балла
Из уравнения: $\mathcal{E}^2R_2/(R_2 + r)^2 = 1,6\mathcal{E}^2R_1/(R_1 + r)^2$	2 балла
получим квадратное уравнение $4r^2 + 6r - 6,75 = 0$.	2 балла
Физический смысл имеет корень $r = 0,75$ Ом.	2 балла
Исследуем зависимость мощности, выделяющейся на резисторе, от силы тока, протекающей через резистор. Напряжение на резисторе с сопротивлением R: $U = \mathcal{E} - Ir$.	2 балла

Мощность, выделяющая на резисторе $P = IU = I(\mathcal{E} - Ir)$,	2 балла
является квадратичной функцией силы тока I. Ветви параболы направлены вниз, поэтому мощность имеет максимум при токе $I_m = (I_1 + I_2) = \mathcal{E}/(2r)$, где	2 балла
$I_1 = 0$, $I_2 = \mathcal{E}/r$ — токи, при которых мощность P равна нулю.	2 балла
Максимум мощности соответствует вершине параболы и равен $P_{max} = P(I_m) = \mathcal{E}^2/(4r) = 6,75$ Вт.	4 балла
Из уравнения $R_3 + r = 2r$ определим сопротивление R_3 резистора, на котором выделяется максимальная мощность $R_3 = r = 0,75$ Ом.	3 балла

Ответ: $r = 0,75$ Ом; $R_3 = 0,75$ Ом; $P_{max} = 6,75$ Вт.