МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. Шухова»**

**(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной

техники и автоматизированных систем

**Лабораторная работа №19.10**

по дисциплине: «Бинарный поиск»

Выполнил/a: ст. группы ВТ-231

Кисиль Николай Владимирович

Проверили:

Черников Сергей Викторович

Новожен Никита Викторович

Белгород, 2023 г.

Цель работы: получение навыков использования алгоритмов бинарного поиска для решения задач оптимизации

Содержание работы

[Задача 1: Двоичный поиск. 3](#_Toc152524589)

[Задача 2: Ближайшее слева 4](#_Toc152524590)

[Задача 3: Ближайшее справа. 5](#_Toc152524591)

[Задача 4: Быстрый поиск в массиве 6](#_Toc152524592)

[Задача 5: Веревочки 7](#_Toc152524593)

[Задача 6: Очень Легкая Задача 8](#_Toc152524594)

[Задача 7: Ночная работа 9](#_Toc152524595)

[Задача 8: Компьютерная игра 10](#_Toc152524596)

[Задача 9: Книги 11](#_Toc152524597)

[Задача 10: Евгений и плейлист 12](#_Toc152524598)

[Задача 11: Алена и узкий холодильник 13](#_Toc152524599)

[Задача 12: Модные числа 14](#_Toc152524600)

[Задача 13: Пара тем 15](#_Toc152524601)

[Задача 14: \* Чемпионат мира 17](#_Toc152524602)

[Задача 15: \* Максимальная медиана 18](#_Toc152524603)

[Задача 16: \*Разделение массива 19](#_Toc152524604)

[Задача 17: \*\* Гамбургеры 21](#_Toc152524605)

[Задача 18: \*\* Slay the Dragon 23](#_Toc152524606)

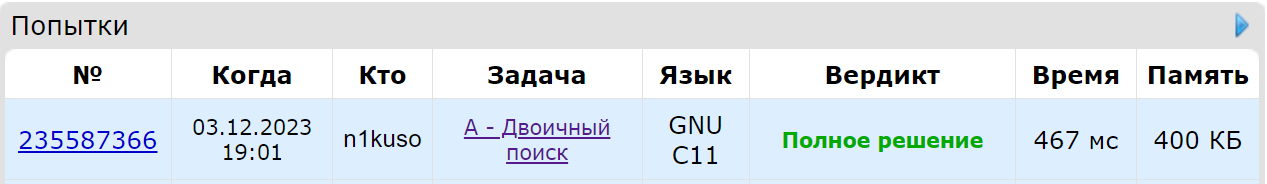
[Задача 20: \*\*Удаление двух элементов 25](#_Toc152524607)

# Задача 1: Двоичный поиск.

Код:

#include <stdio.h>  
#include <stdbool.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
bool find(const int \*a, int n, int x) {  
 int l = 0;  
 int r = n - 1;  
   
 while (r >= l) {  
 int m = (l + r) / 2;  
   
 if (a[m] == x) {  
 return true;  
 } else if (a[m] < x) {  
 l = m + 1;  
 } else {  
 r = m - 1;  
 }  
 }  
 return false;  
}  
  
int main() {  
 int n, k;  
 scanf("%d %d", &n, &k);  
   
 int a[n];  
 inputArray(a, n);  
   
 while (k--) {  
 int x;  
 scanf("%d", &x);  
   
 if (find(a, n, x)) {  
 printf("YES\n");  
 } else {  
 printf("NO\n");  
 }  
 }  
}

Вердикт тестирующей программы:

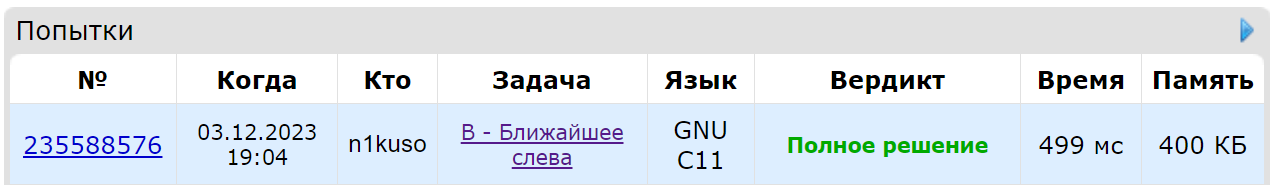


# Задача 2: Ближайшее слева

Код:

#include <stdio.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int find\_lower(const int \* a, int n, int x) {  
 int l = -1;  
 int r = n;  
  
 while (r > l + 1) {  
 int m = (l + r) / 2;  
  
 if(a[m] <= x) {  
 l = m;  
 } else {  
 r = m;  
 }  
 }  
  
 return l + 1;  
}  
  
int main() {  
 int n, k;  
 scanf("%d %d", &n, &k);  
  
 int a[n];  
 inputArray(a, n);  
  
 while(k--) {  
 int x;  
 scanf("%d", &x);  
  
 printf("%d\n", find\_lower(a, n, x));  
 }  
  
 return 0;  
}

Вердикт тестирующей программы:

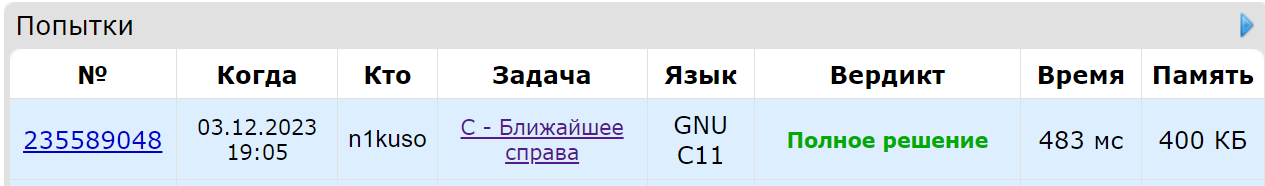


# Задача 3: Ближайшее справа.

Код:

#include <stdio.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int find\_upper(const int \* a, int n, int x) {  
 int l = -1;  
 int r = n;  
  
 while (r > l + 1) {  
 int m = (l + r) / 2;  
  
 if(a[m] < x) {  
 l = m;  
 } else {  
 r = m;  
 }  
 }  
 return r + 1;  
}  
  
int main() {  
 int n, k;  
 scanf("%d %d", &n, &k);  
  
 int a[n];  
 inputArray(a, n);  
  
 while(k--) {  
 int x;  
 scanf("%d", &x);  
  
 printf("%d\n", find\_upper(a, n, x));  
 }  
  
 return 0;  
}

Вердикт тестирующей программы

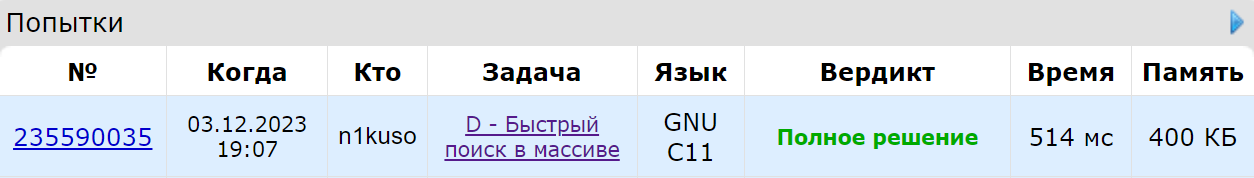


# Задача 4: Быстрый поиск в массиве

Код:

#include <stdio.h>  
#include <stdlib.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int compare(const void \*a, const void \*b) {  
 return (\*(int \*)a - \*(int \*)b);  
}  
  
int find(int \* a, int n, int x) {  
 int l = 0;  
 int r = n - 1;  
   
 while(l <= r) {  
 int m = (l + r) / 2;  
   
 if(a[m] >= x) {  
 r = m - 1;  
 } else {  
 l = m + 1;  
 }  
 }  
   
 return r + 1;  
}  
  
int main() {  
 int n, k;  
 scanf("%d", &n);  
   
 int a[n];  
 inputArray(a, n);  
   
 scanf("%d", &k);  
   
 qsort(a, n, sizeof(int), compare);  
   
 while(k--) {  
 int l, r;  
 scanf("%d %d", &l, &r);  
   
 int dist = find(a, n, r + 1) - find(a, n, l);  
   
 printf("%d\n", dist);  
 }  
}

Вердикт тестирующей программы

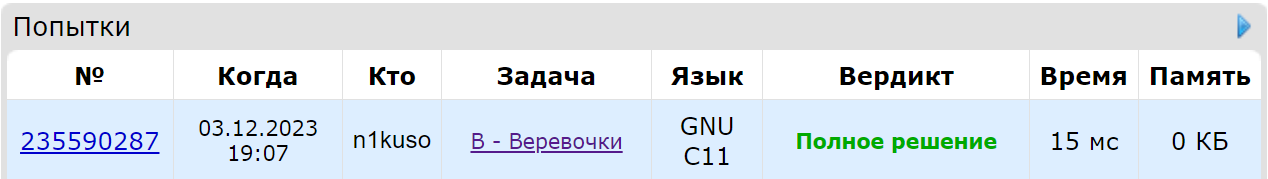


# Задача 5: Веревочки

Код:

#include <stdio.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int main() {  
 int n, k;  
 scanf("%d %d", &n, &k);  
  
 int a[n];  
 inputArray(a, n);  
  
 double l = 0;  
 double r = 1e8;  
  
 for(int i = 0; i < 100; i++) {  
 double m = (l + r) / 2;  
 int s = 0;  
  
 for(int j = 0; j < n; j++) {  
 s += (int)(a[j] / m);  
 }  
 if (s >= k) {  
 l = m;  
 } else {  
 r = m;  
 }  
 }  
  
 printf("%f", r);  
  
 return 0;  
}

Вердикт тестирующей программы

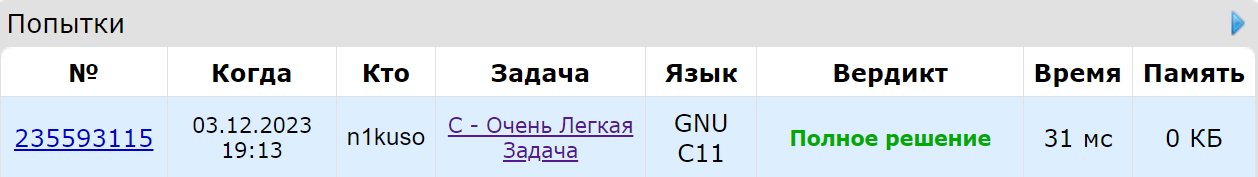


# Задача 6: Очень Легкая Задача

Код:

#include <stdio.h>  
#include <limits.h>  
  
void swap(int \*a, int \*b) {  
 int t = \*a;  
 \*a = \*b;  
 \*b = t;  
}  
  
void sort2(int \*a, int \*b) {  
 if(\*a > \*b) {  
 swap(a, b);  
 }  
}  
  
int countCopies(int t1, int t2, int x) {  
 sort2(&t1, &t2);  
 return x / t1 + (x - t1) / t2;  
}  
  
int binSearchKSER(const int t1, const int t2, const int n) {  
 int l = 0;  
 int r = INT\_MAX;  
  
 while (r - l > 1) {  
 int m = (l + r) / 2;  
 if(countCopies(t1, t2, m) < n) {  
 l = m;  
 } else {  
 r = m;  
 }  
 }  
  
 return r;  
}  
  
int main() {  
 int n, x, y;  
 scanf("%d %d %d", &n, &x, &y);  
  
 printf("%d", binSearchKSER(x, y, n));  
  
 return 0;  
}

Вердикт тестирующей программы

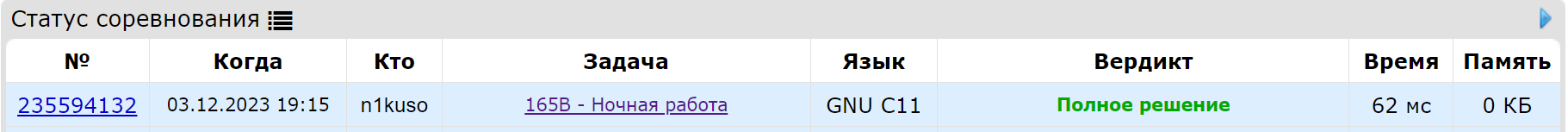


# Задача 7: Ночная работа

Код:

#include <stdio.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int sum(int x, int k) {  
 int sum = 0;  
 while (x) {  
 sum += x;  
 x = x / k;  
 }  
 return sum;  
}  
  
int main() {  
 int n, k;  
 scanf("%d %d", &n, &k);  
  
 int l = 0;  
 int r = n;  
  
 int result = 0;  
 while (l <= r) {  
 int m = (l + r) / 2;  
 int sum\_time = sum(m, k);  
  
 if (sum\_time >= n) {  
 r = m - 1;  
 result = m;  
 } else {  
 l = m + 1;  
 }  
 }  
  
 printf("%d", result);  
 return 0;  
}

Вердикт тестирующей программы

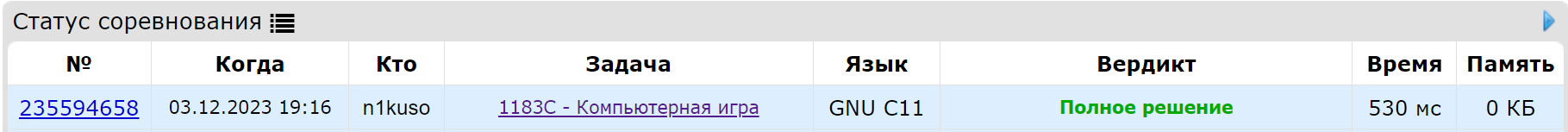


# Задача 8: Компьютерная игра

Код:

#include <stdio.h>  
  
typedef long long ll;  
  
int main() {  
 int q;  
 scanf("%d", &q);  
  
 for(size\_t i = 0; i < q; i++) {  
 ll k, n, a, b;  
 scanf("%lld %lld %lld %lld", &k, &n, &a, &b);  
  
 ll l = 0;  
 ll r = n;  
 ll result = -1;  
  
 while (l <= r) {  
 ll m = (l + r) / 2;  
 ll check = k - ((m \* a) + ((n - m) \* b));  
   
 if (check > 0) {  
 result = m;  
 l = m + 1;  
 } else {  
 r = m - 1;  
 }  
 }  
  
 printf("%lld\n", result);  
 }  
  
 return 0;  
}

Вердикт тестирующей программы

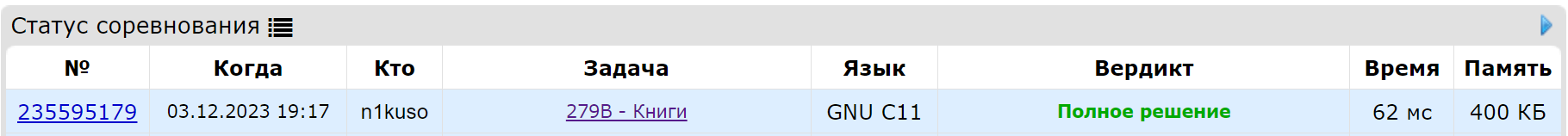


# Задача 9: Книги

Код:

#include <stdio.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int max2(int a, int b) {  
 return a > b ? a : b;  
}  
  
int main() {  
 int n, t;  
 scanf("%d %d", &n, &t);  
   
 int a[n];  
 inputArray(a, n);  
  
 int l = 0;  
 int r = 0;  
 int result = 0;  
  
 while(r < n) {  
 if(a[r] <= t) {  
 t -= a[r];  
 r++;  
 result = max2(result, r - l);  
 } else {  
 t += a[l];  
 l++;  
 }  
 }  
 printf("%d", result);  
 return 0;  
}

Вердикт тестирующей программы

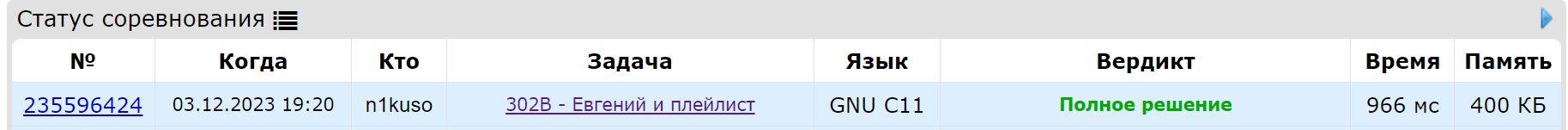


# Задача 10: Евгений и плейлист

Код:

#include <stdio.h>  
  
int binarySearch(const int \* a, const int n, const int x) {  
 int l = 0;  
 int r = n;  
  
 while (l <= r) {  
 int m = (l + r) / 2;  
  
 if (a[m] >= x && a[m - 1] < x) {  
 return m;  
 } else if (a[m] < x) {  
 l = m + 1;  
 } else {  
 r = m - 1;  
 }  
 }  
  
 return 0;  
}  
  
int main() {  
 int n, m;  
 scanf("%d %d", &n, &m);  
  
 int a[n + 1];  
 a[0] = 0;  
  
 for (int i = 0; i < n; i++) {  
 int c, t;  
 scanf("%d %d", &c, &t);  
  
 a[i + 1] = c \* t + a[i];  
 }  
  
 for (int j = 0; j < m; j++) {  
 int v;  
 scanf("%d", &v);  
  
 int number\_song = binarySearch(a, n, v);  
  
 printf("%d\n", number\_song);  
 }  
  
 return 0;  
}

Вердикт тестирующей программы



# Задача 11: Алена и узкий холодильник

Код:

#include <stdio.h>  
#include <stdlib.h>  
  
int comparator(const void \*p, const void \*q) {  
 return (\*(int \*) p - \*(int \*) q);  
}  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
long long getHeight(int \*a, int n) {  
 qsort(a, n, sizeof(int), comparator);  
 long long height = 0;  
 for (int i = n - 1; i >= 0; i -= 2) {  
 height += a[i];  
 }  
  
 return height;  
}  
  
int main() {  
 int n, h;  
 scanf("%d %d", &n, &h);  
  
 int a[n];  
 int b[n];  
 inputArray(a, n);  
  
 int l = 0;  
 int r = n + 1;  
  
 while (r - l > 1) {  
 int m = (r + l) / 2;  
  
 for (int i = 0; i < m; i++) {  
 b[i] = a[i];  
 }  
  
 long long potentialHeight = getHeight(b, m);  
  
 if (potentialHeight <= h) {  
 l = m;  
 } else {  
 r = m;  
 }  
 }  
  
 printf("%d", l);  
  
 return 0;  
}

Вердикт тестирующей программы

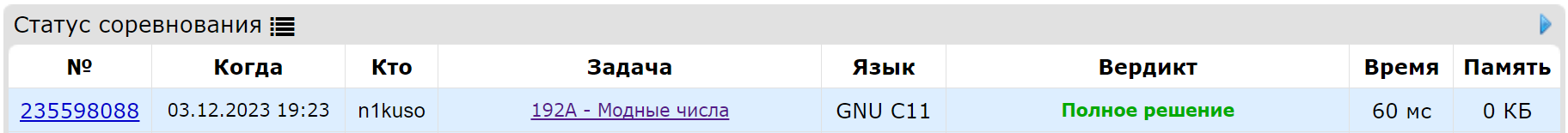


# Задача 12: Модные числа

Код:

#include <stdio.h>  
#include <math.h>  
  
typedef long long ll;  
  
ll getTriangleNum(ll x) {  
 return x \* (x + 1) / 2;  
}  
  
long long binarySearch(int n) {  
 for (int i = 1; i <= sqrt(n) + 1; i++) {  
  
 long long number = getTriangleNum(i);  
 long long l = 1;  
 long long r = n - number + 1;  
  
 while (r - l > 1) {  
 long long m = (l + r) / 2;  
 if (number + getTriangleNum(m) <= n)  
 l = m;  
 else  
 r = m;  
 }  
 if (number + getTriangleNum(l) == n) {  
 return 1;  
 }  
 }  
 return 0;  
}  
  
int main() {  
 int n;  
 scanf("%d", &n);  
  
 if (binarySearch(n) == 0) {  
 printf("NO");  
 } else {  
 printf("YES");  
 }  
  
 return 0;  
}

Вердикт тестирующей программы

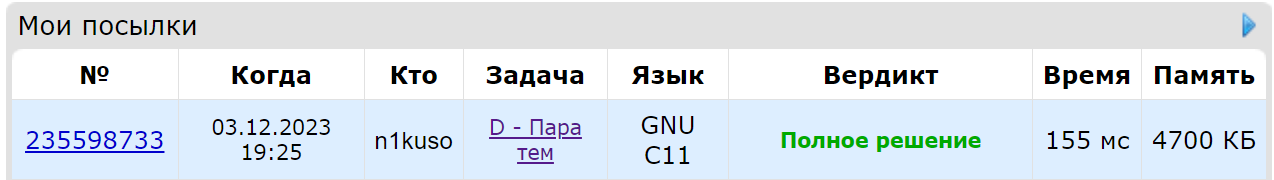


# Задача 13: Пара тем

Код:

#include <stdio.h>  
#include <stdlib.h>  
  
typedef long long ll;  
  
int compare\_ints(const void \*a, const void \*b) {  
 ll arg1 = \*(const ll \*) a;  
 ll arg2 = \*(const ll \*) b;  
  
 if (arg1 < arg2)  
 return -1;  
 if (arg1 > arg2)  
 return 1;  
 return 0;  
}  
  
void inputArray(ll \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%lld", &a[i]);  
}  
  
int binarySearch(const ll \*a, const ll x, const int i) {  
 int l = 0;  
 int r = i;  
  
 while (l <= r) {  
 int m = (l + r) / 2;  
  
 if (a[m] >= -x + 1)  
 r = m - 1;  
 else if (a[m] < -x + 1)  
 l = m + 1;  
 }  
  
 return l;  
}  
  
int main() {  
 int n;  
 scanf("%d", &n);  
  
 ll a[n];  
 inputArray(a, n);  
  
 ll b[n];  
 inputArray(b, n);  
  
 ll c[n];  
 for (int i = 0; i < n; ++i) {  
 c[i] = a[i] - b[i];  
 }  
  
 qsort(c, n, sizeof(ll), compare\_ints);  
  
 ll count = 0;  
 for (int i = 0; i < n; ++i)  
 if (c[i] > 0) {  
 count += i - binarySearch(c, c[i], i);  
 }  
  
 printf("%lld", count);  
  
 return 0;  
}

Вердикт тестирующей программы

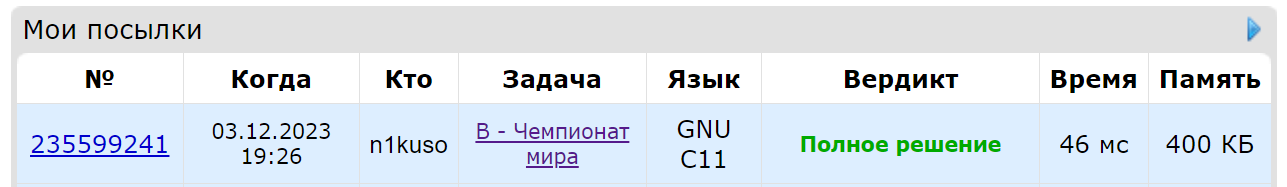


# Задача 14: \* Чемпионат мира

Код:

#include <stdio.h>  
  
void inputArray(int \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%d", &a[i]);  
}  
  
int binarySearch(const int \*a, const int n) {  
 int min\_time = -1;  
 int number = 0;  
  
 for (int i = 0; i < n; i++) {  
 int l = -1;  
 int r = (a[i] - i) / n + 1;  
  
 while (r - l > 1) {  
 int m = (l + r) / 2;  
 if (i + m \* n < a[i]) {  
 l = m;  
 } else {  
 r = m;  
 }  
 }  
   
 int time\_passed = i + r \* n;  
  
 if (time\_passed < min\_time || number == 0) {  
 min\_time = time\_passed;  
 number = i + 1;  
 }  
 }  
  
 return number;  
}  
  
  
int main() {  
 int n;  
 scanf("%d", &n);  
  
 int a[n];  
 inputArray(a, n);  
  
 printf("%d", binarySearch(a, n));  
}

Вердикт тестирующей программы

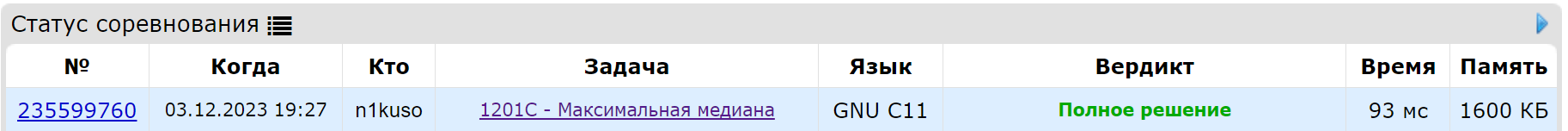


# Задача 15: \* Максимальная медиана

Код:

#include <stdio.h>  
#include <stdlib.h>  
  
typedef long long ll;  
  
void inputArray(ll \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%lld", &a[i]);  
}  
  
int comparator(const void \*a, const void \*b) {  
 return (\*(ll \*) a - \*(ll \*) b);  
}  
  
int isValid(ll \*a, ll x, ll k, ll n) {  
 ll sum = 0;  
 for (int i = n / 2; i < n; i++) {  
 if (x - a[i] > 0) {  
 sum += x - a[i];  
 }  
 }  
 return sum <= k;  
}  
  
int main() {  
 ll n, k;  
 scanf("%lld %lld", &n, &k);  
  
 ll a[n];  
 inputArray(a, n);  
  
 qsort(a, n, sizeof(ll), comparator);  
  
 ll l = 1;  
 ll r = 1e10;  
  
 while (l != r) {  
 ll m = (l + r + 1) / 2;  
  
 if (isValid(a, m, k, n)) {  
 l = m;  
 } else {  
 r = m - 1;  
 }  
 }  
  
 printf("%lld", l);  
   
 return 0;  
}

Вердикт тестирующей программы

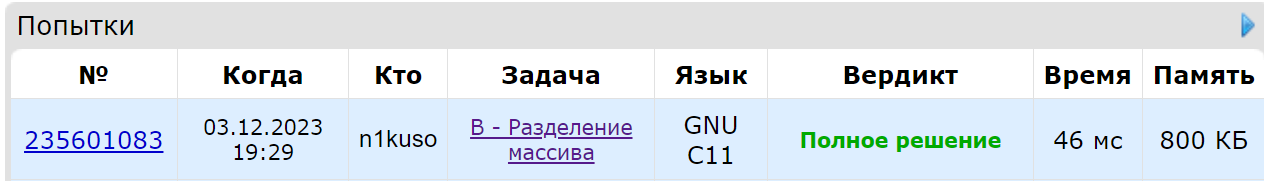


# Задача 16: \*Разделение массива

Код:

#include <stdio.h>  
  
typedef long long ll;  
  
void inputArray(ll \*a, const ll n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%lld", &a[i]);  
}  
  
ll check(ll \*a, ll m, ll n, ll k) {  
 ll parts = 1;  
 ll sum = 0;  
  
 for (ll i = 0; i < n; ++i) {  
 if (a[i] <= m) {  
 sum += a[i];  
 if (sum > m) {  
 parts++;  
 sum = a[i];  
 }  
 } else {  
 return 0;  
 }  
 }  
  
 return parts <= k;  
}  
  
ll binarySearch(ll \*a, ll n, ll k) {  
 ll l = 1;  
 ll r = 1e18;  
 ll result = -1;  
  
 while (l <= r) {  
 ll m = (l + r) / 2;  
  
 if (check(a, m, n, k)) {  
 result = m;  
 r = m - 1;  
 } else {  
 l = m + 1;  
 }  
 }  
  
 return result;  
}  
  
int main() {  
 ll n, k;  
 scanf("%lld %lld", &n, &k);  
  
 ll a[n];  
 inputArray(a, n);  
  
 ll result = binarySearch(a, n, k);  
  
 printf("%lld", result);

Вердикт тестирующей программы

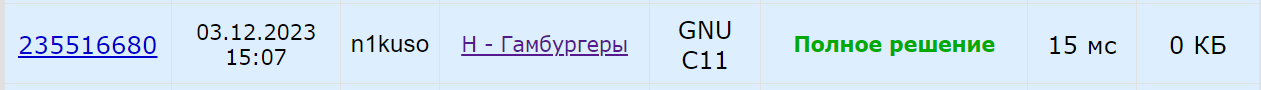


# Задача 17: \*\* Гамбургеры

Код:

#include <stdio.h>  
#include <string.h>  
  
#define MAX\_RECIPE\_LENGTH 100  
  
typedef long long ll;  
  
ll max(ll a, ll b) {  
 return (a > b) ? a : b;  
}  
  
void getRecipe(char \*recipe, int \*nBread, int \*nSausage, int \*nCheese) {  
 int nIngredients = strlen(recipe);  
 \*nBread = 0;  
 \*nSausage = 0;  
 \*nCheese = 0;  
 for (int ingredientIndex = 0; ingredientIndex < nIngredients; ingredientIndex++) {  
 switch (recipe[ingredientIndex]) {  
 case 'B':  
 (\*nBread)++;  
 break;  
 case 'S':  
 (\*nSausage)++;  
 break;  
 case 'C':  
 (\*nCheese)++;  
 break;  
 }  
 }  
}  
  
int main() {  
 char recipe[MAX\_RECIPE\_LENGTH + 1]; // +1 - под ноль-символ  
 int nBread, nSausage, nCheese;  
 gets(recipe);  
 getRecipe(recipe, &nBread, &nSausage, &nCheese);  
  
 ll nb, ns, nc;  
 scanf("%lld %lld %lld", &nb, &ns, &nc);  
 ll pb, ps, pc;  
 scanf("%lld %lld %lld", &pb, &ps, &pc);  
  
 ll rub;  
 scanf("%lld", &rub);  
  
 ll l = 0;  
 ll r = 1e15;  
 ll result = 0;  
  
 while (l <= r) {  
 ll m = (l + r) / 2;  
  
 ll money\_for\_bread = max(0, m \* nBread - nb) \* pb;  
 ll money\_for\_sausage = max(0, m \* nSausage - ns) \* ps;  
 ll money\_for\_cheese = max(0, m \* nCheese - nc) \* pc;  
  
 ll total\_money = money\_for\_bread + money\_for\_sausage + money\_for\_cheese;  
  
 if (total\_money <= rub) {  
 result = m;  
 l = m + 1;  
 } else {  
 r = m - 1;  
 }  
 }  
  
 printf("%lld", result);  
  
 return 0;  
}

Вердикт тестирующей программы



# Задача 18: \*\* Slay the Dragon

Код:

#include <stdio.h>  
#include <stdlib.h>  
  
typedef long long ll;  
  
void inputArray(ll \*a, const int n) {  
 for (size\_t i = 0; i < n; i++)  
 scanf("%lld", &a[i]);  
}  
  
int equate(const void \*first\_number, const void \*second\_number) {  
 ll result = \*(ll \*) first\_number - \*(ll \*) second\_number;  
 return (result > 0) ? 1 : (result == 0) ? 0 : -1;  
}  
  
ll min(ll a, ll b) {  
 return a > b ? b : a;  
}  
  
ll max(ll a, ll b) {  
 return a > b ? a : b;  
}  
  
ll sum\_array(ll \*a, const int n) {  
 ll sum = 0;  
 for (int i = 0; i < n; i++) {  
 sum += a[i];  
 }  
 return sum;  
}  
  
int main() {  
 int n, m;  
 scanf("%d", &n);  
  
 ll a[n];  
 inputArray(a, n);  
  
 qsort(a, n, sizeof(ll), equate);  
  
 ll sum = sum\_array(a, n);  
  
 scanf("%d", &m);  
  
 for (int i = 0; i < m; i++) {  
 ll x, y;  
 scanf("%lld %lld", &x, &y);  
  
 int l = -1;  
 int r = n;  
  
 while (r - l > 1) {  
 int mid = (l + r) / 2;  
 if (a[mid] >= x) {  
 r = mid;  
 } else {  
 l = mid;  
 }  
 }  
  
 ll result = 9e18;  
  
 if (l >= 0) {  
 result = min(result, max(x - a[l], 0) + max(y - sum + a[l], 0));  
 }  
 if (r < n) {  
 result = min(result, max(x - a[r], 0) + max(y - sum + a[r], 0));  
 }  
 printf("%lld\n", result);  
 }  
 return 0;  
}

Вердикт тестирующей программы



# Задача 20: \*\*Удаление двух элементов

Код:

#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
  
#define EPS 1e-6  
  
typedef long long ll;  
  
void inputArray(int \*a, const size\_t n) {  
 for (int i = 0; i < n; ++i)  
 scanf("%d", &a[i]);  
}  
  
  
int compare\_ints(const void \*a, const void \*b) {  
 int arg1 = \*(const int \*) a;  
 int arg2 = \*(const int \*) b;  
  
 if (arg1 < arg2) return -1;  
 if (arg1 > arg2) return 1;  
 return 0;  
}  
  
double arithmeticMean(const int \*a, const size\_t n) {  
 double sum = 0;  
 for (int i = 0; i < n; ++i) {  
 sum += a[i];  
 }  
 return sum / n;  
}  
  
  
ll findFirstPair(const int \*a, const int n, const int i, const double avg) {  
 ll l = i;  
 ll r = n;  
  
 while (r - l > 1) {  
 ll m = (l + r) / 2;  
 if ((double) a[m] + a[i] < avg) {  
 l = m;  
 } else  
 r = m;  
 }  
  
 if (fabs(avg - a[i] - a[r]) < EPS)  
 return r;  
 else  
 return 0;  
}  
  
ll findSecondPair(const int \*a, const int n, const int i, const double avg) {  
 ll l = i + 1;  
 ll r = n;  
  
 while (r - l > 1) {  
 ll m = (l + r) / 2;  
 if ((double) a[m] + a[i] <= avg) {  
 l = m;  
 } else  
 r = m;  
 }  
  
 if (fabs(avg - a[i] - a[l]) < EPS)  
 return l;  
 else  
 return 0;  
}  
  
int main() {  
 int q;  
 scanf("%d", &q);  
  
 for (int i = 0; i < q; i++) {  
 int n;  
 scanf("%d", &n);  
  
 int a[n];  
 inputArray(a, n);  
  
 qsort(a, n, sizeof(int), compare\_ints);  
  
 double avg = arithmeticMean(a, n);  
 ll count = 0;  
  
 for (int i = 0; i < n - 1; i++) {  
 ll first\_number = findFirstPair(a, n, i, 2 \* avg);  
 ll second\_number = findSecondPair(a, n, i, 2 \* avg);  
 if (second\_number != 0 && first\_number != 0) {  
 count += second\_number - first\_number + 1;  
 }  
 }  
  
 printf("%lld\n", count);  
 }  
  
 return 0;  
}

Вердикт тестирующей программы



Вывод: применили навыки написания бинарного поиска для решения задач.