

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 №10**  
по дисциплине: Основы программирования  
тема: «Бинарный поиск»

Выполнил: ст. группы ПВ-223  
Пахомов Владислав Андреевич

Проверили: Притчин Иван Сергеевич  
Черников Сергей Викторович

Код-ревьюер: ст. группы ПВ-223  
Голуцкий Георгий Юрьевич

Белгород 2023 г.

# Лабораторная работа №10

## Вариант 1

### Содержание отчёта:

- Тема лабораторной работы
- Цель лабораторной работы
- Решения задач
  - Название задачи.
  - Исходный код.
  - Вердикт тестирующей системы.
  - Задачи с двумя звёздочками не являются обязательными.
- Работа над ошибками (код-ревью)
- Вывод по работе

**Тема лабораторной работы:** Бинарный поиск

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

## 1. Двоичный поиск

```
#include "stdio.h"
#include "stdlib.h"

int main() {
    int arraySize, requestAmount;
    scanf("%d %d", &arraySize, &requestAmount);

    int *array = (int *) malloc(sizeof(int) * arraySize);
    for (int i = 0; i < arraySize; i++)
        scanf("%d", array + i);

    for (int i = 0; i < requestAmount; i++) {
        int requestedNumber;
        scanf("%d", &requestedNumber);

        int left = -1;
        int right = arraySize;

        while (right - left > 1) {
            int middleIndex = left + (right - left) / 2;
            if (array[middleIndex] < requestedNumber)
                left = middleIndex;
            else
                right = middleIndex;
        }

        if (array[right] == requestedNumber)
            printf("YES\n");
        else
            printf("NO\n");
    }

    free(array);

    return 0;
}
```

[192605327](#)07.02.2023  
16:28

VladOS4052

[А - Двоичный поиск](#)GNU  
C11

Полное решение

452 мс

400 КБ

[Ссылка на репозиторий](#)

## 2. Быстрый поиск в массиве

```
#include "stdio.h"
#include "stdlib.h"

int basicNumComparator(const void *a, const void *b) {
    if (*((int *) a) > *((int *) b)) return 1;
    if (*((int *) a) < *((int *) b)) return -1;
    return 0;
}

int main() {
    int arraySize;
    scanf("%d", &arraySize);

    int *array = (int *) malloc(sizeof(int) * arraySize);
    for (int i = 0; i < arraySize; i++)
        scanf("%d", array + i);

    qsort(array, arraySize, sizeof(array[0]), basicNumComparator);

    int requestAmount;
    scanf("%d", &requestAmount);

    for (int i = 0; i < requestAmount; i++) {
        int requestedLeftBorder, requestedRightBorder;
        scanf("%d %d", &requestedLeftBorder, &requestedRightBorder);

        int left = -1;
        int right = arraySize;

        while (right - left > 1) {
            int middleIndex = left + (right - left) / 2;
            if (array[middleIndex] < requestedLeftBorder)
                left = middleIndex;
            else
                right = middleIndex;
        }

        int leftBorder = right;
        right = arraySize;

        while (right - left > 1) {
            int middleIndex = left + (right - left) / 2;
            if (array[middleIndex] <= requestedRightBorder)
                left = middleIndex;
```

```

        else
            right = middleIndex;
    }

    printf("%d\n", right - leftBorder);
}

free(array);

return 0;
}

```

| №                         | Когда               | Кто        | Задача                                      | Язык       | Вердикт        | Время  | Память |
|---------------------------|---------------------|------------|---|------------|----------------|--------|--------|
| <a href="#">192608765</a> | 07.02.2023<br>16:52 | VladOS4052 | <a href="#">D - Быстрый поиск в массиве</a> | GNU<br>C11 | Полное решение | 529 мс | 400 КБ |

[Ссылка на репозиторий](#)

### 3. Очень Легкая Задача

```
#include "stdio.h"

int printedPages(int x, int y, int time) {
    return time / x + time / y;
}

int main() {
    int copiesAmount, xVelocity, yVelocity;
    scanf("%d %d %d", &copiesAmount, &xVelocity, &yVelocity);

    int left = -1;
    int right = xVelocity * copiesAmount;
    copiesAmount--;

    while (right - left > 1) {
        int middleTime = left + (right - left) / 2;
        if (printedPages(xVelocity, yVelocity, middleTime) < copiesAmount)
            left = middleTime;
        else
            right = middleTime;
    }

    right += xVelocity < yVelocity ? xVelocity : yVelocity;

    printf("%d", right);

    return 0;
}
```

[192725691](#)

08.02.2023 14:26

VladOS4052

[С - Очень Легкая Задача](#)

GNU C11

Полное решение

15 мс

0 КБ

[Ссылка на репозиторий](#)

## 4. Модные числа

```
#include <stdbool.h>
#include "stdio.h"

long long triangleNum(long long num) {
    return num * (num + 1) / 2;
}

int main() {
    long long num;
    scanf("%lld", &num);

    bool result = false;
    for (long long k = 1; (triangleNum(k) < num) && !result; k++) {
        long long secondTriangleNum = num - triangleNum(k);

        long long left = k - 1;
        long long right = num;

        while (right - left > 1) {
            long long middleNum = left + (right - left) / 2;

            if (triangleNum(middleNum) < secondTriangleNum)
                left = middleNum;
            else
                right = middleNum;
        }

        if (secondTriangleNum - triangleNum(right) == 0)
            result = true;
    }

    if (result)
        printf("YES");
    else
        printf("NO");

    return 0;
}
```

| №         | Отправитель                | Задача                    | Язык       | Вердикт        | Время | Память | Отослано               | Протест.               |   |                           |
|-----------|----------------------------|---------------------------|------------|----------------|-------|--------|------------------------|------------------------|---|---------------------------|
| 192732301 | Дорешивание:<br>VladOS4052 | <a href="#">192A</a> - 27 | GNU<br>C11 | Полное решение | 62 мс | 12 КБ  | 2023-02-08<br>15:23:52 | 2023-02-08<br>15:23:52 | ★ | <button>Сравнить</button> |

[Ссылка на репозиторий](#)

## 5. \*\* Чемпионат мира

```
#include <stdio.h>

int timePassed(int gate, int lapsDone, int fanZonesTotal) {
    return gate + lapsDone * fanZonesTotal;
}

int main() {
    int fanZones;
    scanf("%d", &fanZones);

    int minimalTime = -1;
    int gate = 0;
    for (int i = 0; i < fanZones; i++) {
        int order;
        scanf("%d", &order);

        int left = -1;
        int right = (order - i) / fanZones + 1;

        while (right - left > 1) {
            int middle = left + (right - left) / 2;

            if (timePassed(i, middle, fanZones) < order)
                left = middle;
            else
                right = middle;
        }

        int time = timePassed(i, right, fanZones);

        if (time < minimalTime || gate == 0) {
            minimalTime = time;
            gate = i + 1;
        }
    }

    printf("%d", gate);
}
```

[193328243](#)

12.02.2023 13:26

VladOS4052

[В - Чемпионат мира](#)

GNU C11

Полное решение

46 мс

0 КБ

[Ссылка на репозиторий](#)



## 6. \*\* Максимальная медиана

```
#include <stdio.h>
#include <stdlib.h>

int numComparator(const void *a, const void *b) {
    if (*((long long *) a) > *((long long *) b)) return 1;
    if (*((long long *) a) < *((long long *) b)) return -1;
    return 0;
}

long long max(long long a, long long b) {
    return a > b ? a : b;
}

long long countRequiredSum(long long newNumber, long long const *const
↪ numberArray, int arraySize) {
    long long requiredSum = 0;
    for (int i = arraySize / 2; i < arraySize; i++)
        requiredSum += max(0, newNumber - numberArray[i]);

    return requiredSum;
}

int main() {
    int n;
    long long k;
    scanf("%d %lld", &n, &k);

    long long *array = (long long *) malloc(sizeof(long long) * n);
    for (int i = 0; i < n; i++)
        scanf("%lld", array + i);

    qsort(array, n, sizeof(array[0]), numComparator);


    long long left = -1;
    long long right = 2 * 1000000001;

    while (right - left > 1) {
        long long middleNum = left + (right - left) / 2;

        if (countRequiredSum(middleNum, array, n) <= k)
            left = middleNum;
        else
            right = middleNum;
    }
}
```

```
free(array);

printf("%lld", left);
}
```

| №         | Отправитель                | Задача                        | Язык       | Вердикт        | Время  | Память  | Отослано               | Протест.               |   |   |
|-----------|----------------------------|-------------------------------|------------|----------------|--------|---------|------------------------|------------------------|---|---|
| 193328817 | Дорешивание:<br>VladOS4052 | <a href="#">1201C</a> -<br>35 | GNU<br>C11 | Полное решение | 109 мс | 1568 КБ | 2023-02-12<br>13:28:22 | 2023-02-12<br>13:28:22 |  | <input type="button" value="Сравнить"/> |

Ссылка на репозиторий

## 7. \*\* Разделение массива

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

long long max(long long a, long long b) {
    return a > b ? a : b;
}

long long countCuts(long long sectionMaxSum, long long const *const
↪ numberArray, int arraySize) {
    long long tempSum = 0, cuts = 0;

    for (int i = 0; i < arraySize; i++) {
        if (tempSum + numberArray[i] > sectionMaxSum) {
            cuts++;
            tempSum = 0;
        }

        if (numberArray[i] > sectionMaxSum) {
            cuts = -1;
            break;
        }

        tempSum += numberArray[i];
    }

    return cuts;
}

int main() {
    int n;
    long long k;
    scanf("%d %lld", &n, &k);

    long long left = -1;
    long long right = 0;

    long long *array = (long long *) malloc(sizeof(long long) * n);
    for (int i = 0; i < n; i++) {
        scanf("%lld", array + i);
        right += array[i];
    }

    while (right - left > 1) {
```

```
    long long middleNum = left + (right - left) / 2;

    long long cuts = countCuts(middleNum, array, n);
    if (cuts < k && cuts != -1)
        right = middleNum;
    else
        left = middleNum;
}

free(array);

printf("%lld", right);
}
```

|                           |                  |            |  |         |                |       |        |
|---------------------------|------------------|------------|--|---------|----------------|-------|--------|
| <a href="#">193329249</a> | 12.02.2023 13:29 | VladOS4052 | <a href="#">В - Разделение массива</a> | GNU C11 | Полное решение | 46 мс | 800 КБ |
|---------------------------|------------------|------------|--|---------|----------------|-------|--------|

[Ссылка на репозиторий](#)

## 8. \*\* Гамбургеры

```
#include <stdio.h>
#include <string.h>

#define MAX_RECIPE_LENGTH 100

long long max(long long a, long long b) {
    return a > b ? a : b;
}

long long moneyRequire(long long int hamburgerAmount, int breadInHamburger,
↪ int sausageInHamburger, int cheeseInHamburger,
    int breadAmount, int sausageAmount, int cheeseAmount,
    int breadPrice, int sausagePrice, int cheesePrice) {
    return max(hamburgerAmount * breadInHamburger - breadAmount, 0) *
↪ breadPrice +
        max(hamburgerAmount * sausageInHamburger - sausageAmount, 0) *
↪ sausagePrice +
        max(hamburgerAmount * cheeseInHamburger - cheeseAmount, 0) *
↪ cheesePrice;
}

void getRecipe(char *recipe, int *nBread, int *nSausage, int *nCheese) {
    // strlen - функция, определенная в string.h
    // для вычисления длины строки
    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);

    int bAmount, sAmount, cAmount;
    scanf("%d %d %d", &bAmount, &sAmount, &cAmount);

    int bPrice, sPrice, cPrice;
    scanf("%d %d %d", &bPrice, &sPrice, &cPrice);

    long long money;
    scanf("%lld", &money);

    long long left = -1;
    long long right = 1000000000000000;

    while (right - left > 1) {
        long long middle = left + (right - left) / 2;

        if (moneyRequire(middle, nBread, nSausage, nCheese,
                        bAmount, sAmount, cAmount,
                        bPrice, sPrice, cPrice) <= money)
            left = middle;
        else
            right = middle;
    }

    printf("%lld", left);

    return 0;
}

```

|                           |                  |            |                                |         |                |       |      |
|---------------------------|------------------|------------|--------------------------------|---------|----------------|-------|------|
| <a href="#">193189649</a> | 11.02.2023 15:49 | VladOS4052 | <a href="#">H - Гамбургеры</a> | GNU C11 | Полное решение | 15 мс | 0 КБ |
|---------------------------|------------------|------------|--------------------------------|---------|----------------|-------|------|

[Ссылка на репозиторий](#)

## 9. \*\* Slay the Dragon

```
#include <stdio.h>
#include <stdlib.h>

int numComparator(const void *a, const void *b) {
    if (*((long long *) a) > *((long long *) b)) return 1;
    if (*((long long *) a) < *((long long *) b)) return -1;
    return 0;
}

long long max(long long a, long long b) {
    return a > b ? a : b;
}

long long min(long long a, long long b) {
    return a < b ? a : b;
}

int main() {
    long long heroesAmount;
    scanf("%lld", &heroesAmount);

    long long *heroesPower = (long long *) malloc(sizeof(long long) *
↵ heroesAmount);
    long long totalHeroPower = 0;
    for (long long i = 0; i < heroesAmount; i++) {
        scanf("%lld", heroesPower + i);

        totalHeroPower += heroesPower[i];
    }

    qsort(heroesPower, heroesAmount, sizeof(heroesPower[0]),
↵ numComparator);

    int dragonsAmount;
    scanf("%d", &dragonsAmount);

    for (int i = 0; i < dragonsAmount; i++) {
        long long defendingDragonPower, attackingDragonPower;
        scanf("%lld %lld", &defendingDragonPower, &attackingDragonPower);

        long long left = -1;
        long long right = heroesAmount;
        while (right - left > 1) {
            long long middle = left + (right - left) / 2;
```

```

        if (heroesPower[middle] < defendingDragonPower)
            left = middle;
        else
            right = middle;
    }

    long long totalMoneySpent;
    if (left == -1)
        totalMoneySpent = max(0, defendingDragonPower -
↪ heroesPower[right]) +
                                max(0, attackingDragonPower - (totalHeroPower
↪ - heroesPower[right]));
        else if (right == heroesAmount)
            totalMoneySpent = max(0, defendingDragonPower -
↪ heroesPower[left]) +
                                max(0, attackingDragonPower - (totalHeroPower
↪ - heroesPower[left]));
        else
            totalMoneySpent = min(max(0, defendingDragonPower -
↪ heroesPower[right]) +
                                max(0, attackingDragonPower -
↪ (totalHeroPower - heroesPower[right])),
                                max(0, defendingDragonPower -
↪ heroesPower[left]) +
                                max(0, attackingDragonPower -
↪ (totalHeroPower - heroesPower[left])));

    printf("%lld\n", totalMoneySpent);
}

free(heroesPower);

return 0;
}

```

| №         | Отправитель                | Задача                    | Язык       | Вердикт        | Время   | Память  | Отослано               | Протест.               |   |                          |
|-----------|----------------------------|---------------------------|------------|----------------|---------|---------|------------------------|------------------------|---|--------------------------|
| 193329997 | Дорешивание:<br>VladOS4052 | <a href="#">1574C</a> - 9 | GNU<br>C11 | Полное решение | 1231 мс | 1564 КБ | 2023-02-12<br>13:31:29 | 2023-02-12<br>13:31:29 | ★ | <a href="#">Сравнить</a> |

[Ссылка на репозиторий](#)



## 10. \*\* Детский праздник

```
#include <stdio.h>
#include <stdlib.h>

long long min(long long a, long long b) {
    return a < b ? a : b;
}

long long max(long long a, long long b) {
    return a > b ? a : b;
}

typedef struct Worker {
    long long balloonMakingTime;
    long long balloonAmount;
    long long relaxTime;
} Worker;

typedef struct Report {
    long long *workersReport;
    long long workersAmount;
    long long ballonsTotal;
} Report;

Report balloons(long long time, Worker *workers, int workersAmount) {
    Report report = {(long long *) malloc(sizeof(long long) *
↪ workersAmount), workersAmount, 0};

    for (int i = 0; i < workersAmount; i++) {
        Worker worker = workers[i];
        long long fullCycleTime = worker.balloonMakingTime *
↪ worker.balloonAmount + worker.relaxTime;
        long long balloonsFullCycle = (time / fullCycleTime) *
↪ worker.balloonAmount;
        long long timeLeft = max(0, time - (fullCycleTime *
↪ balloonsFullCycle / worker.balloonAmount));
        balloonsFullCycle += min(timeLeft, worker.balloonMakingTime *
↪ worker.balloonAmount) /
            worker.balloonMakingTime;
        report.ballonsTotal += balloonsFullCycle;
        report.workersReport[i] = balloonsFullCycle;
    }

    return report;
}
```

```

int main() {
    long long balloonsAmount;
    int workersAmount;
    scanf("%lld %d", &balloonsAmount, &workersAmount);

    Worker *workers = (Worker *) malloc(sizeof(Worker) * workersAmount);
    for (int i = 0; i < workersAmount; i++) {
        long long balloonMakingTime, balloonAmount, relaxTime;
        scanf("%lld %lld %lld", &balloonMakingTime, &balloonAmount,
↵ &relaxTime);
        workers[i] = (Worker) {balloonMakingTime, balloonAmount,
↵ relaxTime};
    }

    long long left = -1;
    long long right = 17000000000;
    Report fReport = {NULL, 0, 0};

    while (right - left > 1) {
        long long middle = left + (right - left) / 2;
        Report cReport = balloons(middle, workers, workersAmount);
        if (cReport.ballonsTotal < balloonsAmount) {
            left = middle;
            free(cReport.workersReport);
        } else {
            free(fReport.workersReport);
            fReport = cReport;
            right = middle;
        }
    }

    printf("%lld\n", right);

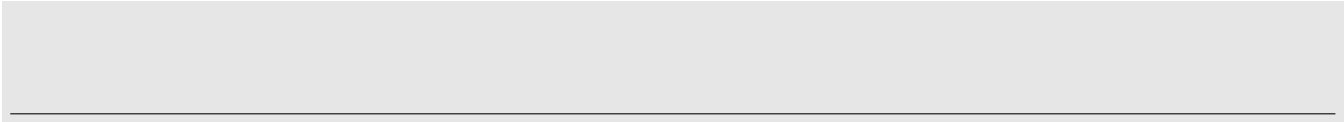
    for (int i = 0; i < workersAmount; i++) {
        printf("%lld ", min(balloonsAmount, fReport.workersReport[i]));

        balloonsAmount -= fReport.workersReport[i];
        balloonsAmount = balloonsAmount < 0 ? 0 : balloonsAmount;
    }

    free(workers);

    return 0;
}

```



|                           |                  |            |                                      |         |                |       |        |
|---------------------------|------------------|------------|--------------------------------------|---------|----------------|-------|--------|
| <a href="#">193334886</a> | 12.02.2023 14:14 | VladOS4052 | <a href="#">D - Детский праздник</a> | GNU C11 | Полное решение | 15 мс | 100 КБ |
|---------------------------|------------------|------------|--------------------------------------|---------|----------------|-------|--------|

Ссылка на репозиторий

**Вывод:** в ходе лабораторной работы получили навыки использования алгоритмов бинарного поиска для решения задач оптимизации.