

Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ
З ПРАКТИЧНОЇ РОБОТИ №3
дисципліна: «Алгоритми та структур и даних»

Виконала: студентка групи КС-22

Узенкова Дар'я

Перевірив: Олешко Олег

Харків

2024

Завдання. На прямій дощечці вбиті гвіздки. Будь-які два гвіздки можна з'єднати ниточкою. Потрібно з'єднати деякі пари гвіздків ниточками так, щоб до кожного гвіздка була прив'язана хоча б одна ниточка, а сумарна довжина всіх ниточок була мінімальна.

Вхідні дані: У першому рядку вхідного потоку записане число N - кількість гвіздків ($2 \leq N \leq 100$). У наступному рядку записано N чисел - координати всіх гвіздків (невід'ємні цілі числа, що не перевершують 10000).

Вихідні дані: мінімальна сумарна довжина всіх ниточок

Результати виконання завдання 1 наведено:

1. У лістингу 1 – вихідний код програми
2. У малюнках 1, 2, 3 – результати виконання програми.

```
#include <stdio.h>

// Функція сортування пузирком по зростанню
int bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // Міняємо елементи місцями
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
            else if (arr[j] == arr[j+1]) return 0;
        }
    }
    return 1;
}

int main() {
    int N;
```

```

printf("Введіть кількість цвяхів: ");
scanf("%d", &N);

int a[N];
int d[N];

printf("Введіть координати цвяхів: ");
for (int i = 0; i < N; i++) {
    scanf("%d", &a[i]);
}

// Сортуюємо масив координат за зростанням
int result = bubbleSort(a, N);
if (!result) return printf("Не може бути цвяхів з однаковими
координатами.\n");

// Ініціалізація початкових значень
d[0] = 0;
d[1] = a[1] - a[0];
d[N-2] = 0;

// Обчислюємо мінімальні відстані для всіх цвяхів
for (int i = 2; i < N - 2; i++) {
    int distanceToNext = a[i + 1] - a[i];
    int distanceToPrev = a[i] - a[i - 1];

    if (distanceToNext < distanceToPrev || distanceToNext ==
distanceToPrev && d[i-1]) {
        d[i] = 0;
        d[i + 1] = distanceToNext;
    }
}

```

```

        else if (distanceToNext > distanceToPrev || distanceToNext ==
distanceToPrev && !d[i-1]){
            d[i] = distanceToPrev;
        }
    }

    d[N-1] = a[N-1] - a[N-2];

    int sum = 0;
    for (int i = 0; i < N; i++) {
        sum += d[i];
    }

    printf("Мінімальна сумарна довжина ниточок: %d\n", sum);

    return 0;
}

```

Лістинг 1 – вихідний код програми

```

Введіть кількість цвяхів: 6
Введіть координати цвяхів: 3 4 12 6 13 14
Мінімальна сумарна довжина ниточок: 5

-----
Process exited after 9.454 seconds with return value 0
Press any key to continue . . .

```

Малюнок 1 – результат виконання програми

```

Введіть кількість цвяхів: 5
Введіть координати цвяхів: 0 2 4 10 12
Мінімальна сумарна довжина ниточок: 6

-----
Process exited after 5.407 seconds with return value 0
Press any key to continue . . . █

```

Малюнок 2 – результат виконання програми

```
Введіть кількість цвяхів: 5
Введіть координати цвяхів: 2 4 6 7 2
Не може бути цвяхів з однаковими координатами.

-----
Process exited after 6.313 seconds with return value 86
Press any key to continue . . .
```

Малюнок 3 – результат виконання програми