Міністерство освіти і науки України

Національний університет «Запорізька політехніка»

кафедра програмних засобів

ЗВІТ

з лабораторної роботи №4

з дисципліни «**Спортивне програмування**» на тему:

**«**РЕКУРЕНТНІ ПОСЛІДОВНОСТІ»

Виконав:

ст. гр. КНТ-113сп Олексій СІМІЧОВ

Прийняв:

ст. викл.: Сергій ЛЕОЩЕНКО

2024

# Мета роботи

1.1.1 Вивчити основні можливості та принципи роботи із жадібними алгоритмами.

# Завдання до лабораторної роботи

2.2.9 Враховуючи список дій із зазначенням часу їх початку та закінчення, виберіть максимальну кількість дій, які може виконувати одна людина, припускаючи, що людина може працювати лише над однією дією одночасно.

Вхідні дані: [(1, 2), (3, 4), (0, 6), (5, 7), (8, 9), (5, 9)]

Вихідні дані: [(1, 2), (3, 4), (5, 7), (8, 9)]

2.2.10 Враховуючи набір завдань, де кожне завдання має кінцевий термін і пов'язаний з ним прибуток, нам потрібно максимізувати прибуток таким чином, щоб кожне завдання було виконано у встановлені терміни.

Вхідні дані:

−масив Завдання: [{дедлайн: 2, прибуток: 100}, {дедлайн: 1, прибуток: 50}, {дедлайн: 2, прибуток: 10}, {дедлайн: 1, прибуток: 20}];

−кількість доступних часових інтервалів: 2.

Вихідні дані: Максимальний прибуток: 150

# 3. Текст розробленої програми

Задача 9:

function greedyActivitySelection(activities) {

  const sortedActivities = activities.sort((a, b) => a[1] - b[1]);

  const selectedActivities = [];

  let lastEndTime = null;

  sortedActivities.forEach((activity) => {

    const [start, end] = activity;

    if (lastEndTime === null || start >= lastEndTime) {

      selectedActivities.push(activity);

      lastEndTime = end;

    }

  });

  return selectedActivities;

}

const activities = [

  [1, 2],

  [3, 4],

  [0, 6],

  [5, 7],

  [8, 9],

  [5, 9],

];

const selected = greedyActivitySelection(activities);

console.log('Вихідні дані:', selected);

Задача 10:

function maximizeProfit(tasks, intervals) {

  tasks.sort((a, b) => b['прибуток'] - a['прибуток']);

  let intervalAvailable = Array(intervals).fill(true);

  let totalProfit = 0;

  tasks.forEach((task) => {

    const deadline = task['дедлайн'];

    const profit = task['прибуток'];

    for (let i = Math.min(deadline, intervals) - 1; i >= 0; i--) {

      if (intervalAvailable[i]) {

        intervalAvailable[i] = false;

        totalProfit += profit;

        break;

      }

    }

  });

  return totalProfit;

}

const tasks = [

  { дедлайн: 2, прибуток: 100 },

  { дедлайн: 1, прибуток: 50 },

  { дедлайн: 2, прибуток: 10 },

  { дедлайн: 1, прибуток: 20 },

];

const intervals = 2;

const maxProfit = maximizeProfit(tasks, intervals);

console.log('Максимальний прибуток:', maxProfit);

# 4. Копії екрану, що відображають результати виконання лабораторної роботи.

Задача 9:



Задача 10:



# 5. Висновки

Під час виконання лабораторної роботи, я вивчив основні можливості та принципи роботи із жадібними алгоритмами.