Міністерство освіти і науки України

Національний університет «Запорізька політехніка»

кафедра програмних засобів

ЗВІТ

з лабораторної роботи №4

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

**«**ЖАДІБНІ АЛГОРИТМИ»

Виконали:

ст. гр. КНТ-113сп Ярослав ПАНЧЕНКО  
Михайло ГРИГОРОВ

Прийняв:

ст.викл. Сергій ЛЕОЩЕНКО

2023

# Мета роботи

Вивчити основні можливості та принципи роботи із жадібними алгоритмами.

# Завдання до лабораторної роботи

3.1 Ознайомитися з основними теоретичними відомостями за темою роботи, використовуючи ці методичні вказівки, а також рекомендовану літературу.

3.2 Обрати та виконати дві задачі із запропонованого переліку.

3.2.9 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)]

Пояснення: максимальна кількість дій, які можуть бути виконані, дорівнює 4, а

обраними діями є (1, 2), (3, 4), (5, 7), і (8, 9).

3.2.10 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

Пояснення: ми можемо вибрати перше і третє завдання, які приносять Загальний прибуток у розмірі 150 і відповідають термінам виконання.

# Текст розробленої програми.

3.2.9

function maxNonOverlappingActivities(activities) {

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

  const selectedActivities = [];

  let lastEndTime = -1;

  for (const activity of activities) {

      const [startTime, endTime] = activity;

      if (startTime >= lastEndTime) {

          selectedActivities.push(activity);

          lastEndTime = endTime;

      }

  }

  return selectedActivities;

}

const activities = [

  [1, 2],

  [3, 4],

  [0, 6],

  [5, 7],

  [8, 9],

  [5, 9]

];

const result = maxNonOverlappingActivities(activities);

console.log("Обрані дії:", result);

3.2.10

function maximizeProfit(tasks, deadline) {

  tasks.sort((a, b) => b.profit - a.profit);

  let maxProfit = 0;

  const schedule = [];

  for (let i = 0; i < tasks.length; i++) {

    const task = tasks[i];

    if (task.days <= deadline) {

      schedule.push(task);

      maxProfit += task.profit;

      deadline -= task.days;

    }

  }

  return { maxProfit, schedule };

}

const tasks = [

  { days: 3, profit: 100 },

  { days: 1, profit: 50 },

  { days: 2, profit: 60 },

  { days: 2, profit: 40 },

  { days: 1, profit: 30 },

];

const deadline = 2;

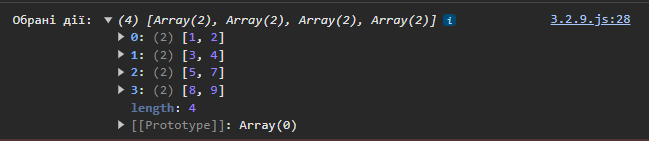
const result = maximizeProfit(tasks, deadline);

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

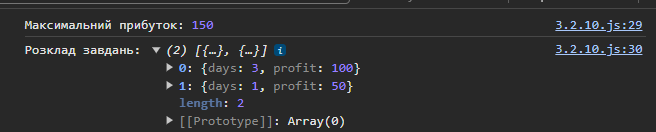
console.log('Розклад завдань:', result.schedule);

# Копії екрану, що відображають результати виконання лабораторної роботи.

3.2.9



3.2.10



# 5. Висновки

Вивчив основні можливості та принципи роботи із жадібними алгоритмами.