

## Изпит по Функционално програмиране

спец. Компютърни науки, 19.2.2013 г.

Вариант А

**Задача 1.** Да се реализира функция *maxDistance*, която по даден списък от точки представени като двойки реални числа - координати в равнината намира максималното разстояние между кои да е две точки.

Пример: *maxDistance* [(-1.1, 1), (1.8, 2), (3, 1), (-1, -2)] → 5

**Задача 2.** Да се реализира функция *selectionSort*, която прави сортиране на списък от числа по метода на пряката селекция, без да използва функции за пряк достъп до произволен елемент на списък.

Пример: *selectionSort* [5, 4.4, 3, 6, 1] → [1, 3, 4.4, 5, 6]

**Задача 3.** Даден е ориентиран граф без цикли. Във върховете на графа са записани цели числа. Дефинирайте предикат *isReachable*, който по даден граф и върхове в него *v1* и *v2* проверява дали има път от *v1* до *v2* такъв, че започвайки с начална сума от 0 и прибавяйки към нея стойностите на всеки връх, през който минава пътя (включително *v1* и *v2*), сумата никога не става отрицателна.

Опишете накратко представянето на графа, което използвате.

**Задача 4.** Да се реализира функция *matchedBy*, която по даден регулярен израз, съдържащ единствено латински букви и точно един символ \*, означаващ 0 или повече срещания на предходния символ, връща поток от думите, разпознавани от този регулярен израз.

Пример: *matchedBy* "Tro\*l" → ["Trl", "Trol", "Trool", ...]

---

## Изпит по Функционално програмиране

спец. Компютърни науки, 19.2.2013 г.

Вариант Б

**Задача 1.** Да се реализира функция *nCloserThanN*, която по даден списък от точки представени като двойки реални числа - координати в равнината и число *n* намира броя на двойките различни точки, които са на разстояние по-малко от *n*. Двойките се считат за ненаредени, т.е.  $((x_1, y_1), (x_2, y_2))$  е същата като  $((x_2, y_2), (x_1, y_1))$ .

Пример: *nCloserThanX* [(-1.1, 1), (1.8, 2), (3, 1), (-1, -2)] 4 → 3

**Задача 2.** Да се реализира функция *mergeSort*, която прави сортиране на списък от числа по метода на сливането, без да използва функции за пряк достъп до произволен елемент на списък.

Пример: *mergeSort* [5, 4.4, 3, 6, 1] → [1, 3, 4.4, 5, 6]

**Задача 3.** Даден е ориентиран граф без цикли. Във върховете на графа са записани цели числа. Дефинирайте предикат *isSumReachable*, който по даден граф и връх в него *v1* и цяло число *n* проверява дали има път от *v1* такъв, че започвайки с начална сума от 0 и прибавяйки към нея стойностите на всеки връх, през който минава пътя (включително *v1*), сумата става равна точно на *n*.

Опишете накратко представянето на графа, което използвате.

**Задача 4.** Да се реализира функция *matchedBy*, която по даден регулярен израз, съдържащ единствено латински букви и точно един символ +, означаващ 1 или повече срещания на предходния символ, връща поток от думите, разпознавани от този регулярен израз.

Пример: *matchedBy* "tl+DR" → ["tIDR", "tIIDR", "tIIIDR", ...]