

Algorytmy i Struktury Danych – Kolejki

Laboratorium nr 3 i 4 (4 x 45") / 18.11.2021

Grupa WCY20IY2S1

Temat:

a) Kolejki LIFO, FIFO i priorytetowe - implementacja,

Zad. nr 1 (4 x 45" – t. ćw.)

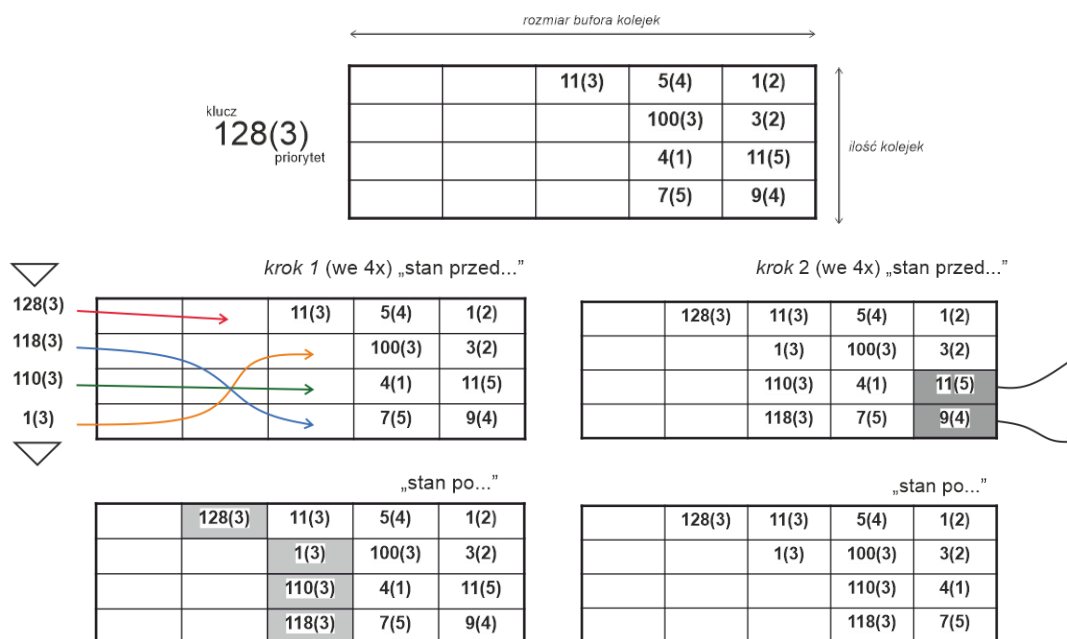
Bazując na strukturze listy dwukierunkowej wykonaj implementację kilku kolejek priorytetowych.

W programie powinny zostać zaimplementowane przedstawione funkcje:

- **addQuel**(q, (x, key)) – dodanie do kolejki *q* elementu *x* wraz z priorytetem *key*,
- **extractQuel**(q) – zdjęcie elementu z kolejki *q* o najwyższym priorytecie (*key*),
- **isEmpty**(q) – sprawdzenie stanu kolejki *q*,
- **sizeQuel**(q) – ilość elementów oczekujących w kolejce *q*,
- **printQuel**(q) – wypisz zawartość kolejki *q* wraz z komunikatami (jeżeli są konieczne).
- **maxQ**(q) – wypisuje element o największym kluczu w kolejce *q*,
- **incKey**(q, x) – zmienia wartość klucza *key* elementu *x*.

Założenia:

- system obsługi powinien zawierać cztery kolejki priorytetowe,
- długości kolejek są identyczne i równe **n** /inicjowana zmienna programu/,
- kolejki są inicjowane losową liczbą elementów: *x(key)* (nie większą niż **4n**),
- elementy do kolejki wchodzi **2 razy szybciej** niż **wychodzą** (należy to uwzględnić podczas programowania cyklu obsługi). W jednym cyklu do kolejek wchodzi cztery elementy a w kolejnym wychodzą dwa (zobacz rysunek).
- po przepełnieniu buforów wszystkich kolejek system czeka na obsługę wypełnionych kolejek i nie przyjmuje „nowych” elementów,
 - sygnalizacja stanu kolejki oczekującej: **WAIT**,
 - sygnalizacja stanu kolejki pustej: **EMPTY**,
- element przychodzący powinien trafiać na kolejkę, która zawiera najmniej elementów,
- element przychodzący powinien mieć zdefiniowany priorytet (losowy: liczba z zakresu <1,n>) oraz wartość, klucz (losową: liczba z zakresu <1,100>),
- system powinien pozwolić **na krokową analizę zawartości kolejek**. (cykl: elementy wchodzące, wypis zawartości kolejek, elementy wychodzące; zawartości kolejek proszę przedstawić w wierszach); symulacja obejmuje domyślnie nieskończoną liczbę cykli.
- system powinien działać na strukturze list jednokierunkowych (n list jednokierunkowych).



Rys. 1 Inicjacja, wstawienie oraz obsługa kolejek (n=5). Schemat.

Uwagi dodatkowe:

- Kod programu powinien mieścić się w jednym pliku (włącznie z funkcjami). Przedstawione rozwiązanie powinno być wykonane samodzielnie. W przypadku stwierdzenia kopii wszyscy autorzy identycznych rozwiązań otrzymują ocenę ndst., co wiąże się z konieczności poprawy ćwiczenia w terminie poprawkowym.
- Do implementacji proszę użyć języka C w standardzie ANSI C lub C++ (bez typów generycznych i bibliotek specjalistycznych)
- Z ćwiczenia laboratoryjnego piszemy sprawozdanie w formie elektronicznej, które jest dodawane do ćwiczenia w oddzielnym pliku w formacie docx, pdf, odt. W sprawozdaniu zamieszczamy odpowiedzi na pytania opisowe i niezbędne obliczenia.
- Gotowe rozwiązania (sprawozdanie i kod programu – dwa pliki niespakowane) przesyłamy za pomocą narzędzia MS Teams do oceny. Ocena po sprawdzeniu będzie dostępna dla każdego studenta w systemie MS Teams.