

1. Demonstrati corectitudinea sortarii prin selectie folosind invarianti la ciclare.
2. Considerand un min-heap binar, cu operatiile *insert* si *delete_min*, se cere:
 - a. Care este complexitatea celor doua operatii in cazul cel mai defavorabil ? Justificati.
 - b. Definiti o functie potential pentru min-heap astfel incat costurile amortizate ale celor doua operatii sa fie cat mai mici (*delete_min* sa fie $O(1)$).
 - c. Demonstrati prin metoda potentialului care sunt costurile amortizate ale celor doua operatii.
3. Se considera o secventa de n operatii ce se executa pe o structura de date. Operatia i are costul i daca i este o putere a lui 2, respectiv, are costul egal cu 1 altfel. Determinati costul amortizat al unei operatii folosind atat metoda amortizata, cat si metoda potentialului.

Pentru tipul LIST definit prin constructorii:

$[] : \rightarrow \text{LIST}$

$[a] : T \rightarrow \text{LIST}$

$a::x$ (sau $\text{cons}(a, x)$) : $T \times \text{LIST} \rightarrow \text{LIST}$ (adica a adaugat la inceputul listei)

Folosind recursivitatea si inductia structurala, rezolvati urmatoarele exercitii:

4. Definiti functiile:

- $\text{append}(l_1, l_2)$ – concatenarea a doua liste
- $\text{member}(a, l)$ – verifica apartenenta elementului a in lista l
- $\text{size}(l)$ – dimensiunea listei
- $\text{maxelem}(l)$, $\text{minelem}(l)$ – intorc elementul maxim, respectiv minim din lista
- $\text{set}(l)$ – intoarce lista ce contine elemente distincte din l
- $\text{nbrelems}(l)$ – numarul de elemente distincte din lista

5. Aratati ca orice l de tip LIST:

- $\text{member}(a, l) \rightarrow \text{minelem}(l) \leq a \leq \text{maxelem}(l)$

6.

- $\text{size}(\text{set}(l)) \leq \text{size}(l)$
- $\text{nbrelems}(l) \leq \text{size}(l)$

7.

- $\text{size}(\text{set}(l)) = \text{size}(l) \rightarrow \text{set}(l) = l$

8. Definiti functiile:

$\text{remove}(l, a) : \text{LIST} \times T \rightarrow \text{LIST}$ - intoarce o lista rezultata din eliminarea elementelor a din l

$\text{double}(l, a) : \text{LIST} \times T \rightarrow \text{LIST}$ - intoarce o lista rezultat din dublarea elementelor a din l (fiecare a devine aa)

Sa se demonstreze ca $\text{member}(a, l) \rightarrow \text{size}(\text{remove}(l, a)) \neq \text{size}(\text{double}(l, a))$

9. Tot pentru tipul LIST, implementati algoritmul sortarii prin insertie. Pentru aceasta, implementati axioma / operatia $\text{isort} : \text{LIST} \rightarrow \text{LIST}$ (care primeste o lista si intoarce lista sortata), precum si alte axiome / operatii (mai mici) pe care le considerati utile pentru definirea isort . De asemenea, demonstrati prin inductie structurala ca isort functioneaza corect (pentru aceasta puteti construi o axioma $\text{is_sorted} : \text{LIST} \rightarrow \text{bool}$ care intoarce 1 daca lista este sortata).

Notare: Fiecare exercitiu valoreaza 1p $\Rightarrow 9 \times 1p + 1p$ (oficiu) = 10p