

Prioritásos sor típus

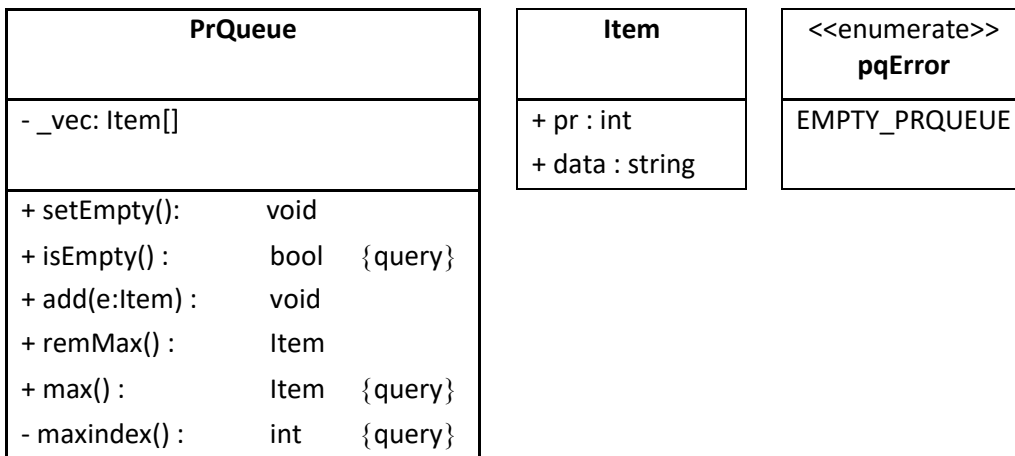
Prioritásos sor

| | | | | | | | | | | | | | | | | | | | |
|--|---|------------------|--|------|--|----------------------------|-------------------------------------|---------------|------------------------|----------------------|--------------------------|--|------------------|--|------|--|----------------------------|-------------------------------------|---------------|
| <p>Típus értékek:</p> <p>PrQueue a maximum prioritásos sorok halmaza, amely soroknak az elemei $\mathbb{Z} \times \mathbb{S}$ típusú párok.</p> | <p>Típus műveletek:</p> <p>pq.setEmpty() pq : PrQueue <i>// Kiüríti a pr sort</i></p> <p>l := pq.isEmpty() pq : PrQueue, l : \mathbb{L} <i>//Igazat ad, ha üres a pr sor, hamisat ha nem.</i></p> <p>pq.add(e) pq : PrQueue, e : $\mathbb{Z} \times \mathbb{S}$ <i>//Beteszi egy új elemet a pr sorba.</i></p> <p>e := pq.remMax() pq : PrQueue, e : $\mathbb{Z} \times \mathbb{S}$ <i>//Kiveszi az egyik legnagyobb prioritású elemet. Fontos, hogy a sor nem lehet üres.</i></p> <p>e := pq.max() pq : PrQueue, e : $\mathbb{Z} \times \mathbb{S}$ <i>//Visszadja az egyik legnagyobb prioritású elemet, nem veszi ki. Fontos, hogy a sor nem lehet üres.</i></p> | | | | | | | | | | | | | | | | | | |
| <p>Típus reprezentáció:</p> <p>vec: Item[] – prioritásos sor elemeit rendezetlen tároló tömb, ahol Item = rec(pr : \mathbb{Z}, data : \mathbb{S})</p> | <p>Típusműveletek implementációja:</p> <p>pq.setEmpty() vec := <> (C++ vector: clear())</p> <p>l := pq.isEmpty() l := vec = 0 (C++ vector: size())</p> <p>pq.add(e) vec := vec \oplus <e> (C++ vector: push_back())</p> <p>e, err := pq.remMax()</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">err := (vec =0)</td> </tr> <tr> <td colspan="2" style="text-align: center;">-err</td> </tr> <tr> <td style="width: 70%;">max, ind := maxSearch(vec)</td> <td rowspan="4" style="text-align: center; vertical-align: middle;">Hiba: üres a prioritásos sor</td> </tr> <tr> <td>e := vec[ind]</td> </tr> <tr> <td>vec[ind] := vec[vec]</td> </tr> <tr> <td>vec:=vec[1.. vec -1]</td> </tr> <tr> <td colspan="2" style="text-align: center;">(C++ vector: pop_back())</td> </tr> </table> <p>e, err := pq.max()</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">err := (vec =0)</td> </tr> <tr> <td colspan="2" style="text-align: center;">-err</td> </tr> <tr> <td style="width: 70%;">max, ind := maxSearch(vec)</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">Hiba: üres a prioritásos sor</td> </tr> <tr> <td>e := vec[ind]</td> </tr> </table> | err := (vec =0) | | -err | | max, ind := maxSearch(vec) | Hiba: üres a prioritásos sor | e := vec[ind] | vec[ind] := vec[vec] | vec:=vec[1.. vec -1] | (C++ vector: pop_back()) | | err := (vec =0) | | -err | | max, ind := maxSearch(vec) | Hiba: üres a prioritásos sor | e := vec[ind] |
| err := (vec =0) | | | | | | | | | | | | | | | | | | | |
| -err | | | | | | | | | | | | | | | | | | | |
| max, ind := maxSearch(vec) | Hiba: üres a prioritásos sor | | | | | | | | | | | | | | | | | | |
| e := vec[ind] | | | | | | | | | | | | | | | | | | | |
| vec[ind] := vec[vec] | | | | | | | | | | | | | | | | | | | |
| vec:=vec[1.. vec -1] | | | | | | | | | | | | | | | | | | | |
| (C++ vector: pop_back()) | | | | | | | | | | | | | | | | | | | |
| err := (vec =0) | | | | | | | | | | | | | | | | | | | |
| -err | | | | | | | | | | | | | | | | | | | |
| max, ind := maxSearch(vec) | Hiba: üres a prioritásos sor | | | | | | | | | | | | | | | | | | |
| e := vec[ind] | | | | | | | | | | | | | | | | | | | |

Több művelet is használja a $(\text{max}, \text{ind}) = \text{MAX}_{i=1..|\text{vec}|} (\text{vec}[i].\text{pr})$ maximum kiválasztást.

Prioritásos sor típus

UML diagrammok:



Megjegyzés: maxindex() a maxSearch() azon változata, mely csak a maximum indexét határozza meg.

Egység teszt példa:

| remMax() tesztelése | | | |
|--|------------|----------------------------|--|
| teszteset | teszt tömb | eredmény | tömb új tartalma |
| üres intervallum | <> | hiba (kivétel dobás) | <> |
| egy elemű | <3> | 3 | <> |
| több elemű esetek: | | | |
| első a legnagyobb | <5,2,3> | 5 | <3,2> |
| utolsó a legnagyobb | <1,2,3> | 3 | <1,2> |
| belső a legnagyobb | <1,3,2> | 3 | <1,2> |
| nem egyértelmű, első és utolsó a legnagyobb | <5,2,5'> | 5 | <5',2> (az adat rész segítségével ellenőrizhető, hogy az elsőt vettük ki) |
| nem egyértelmű, belső és utolsó a legnagyobb | <1,3,3'> | 3 | <1,3'> |
| mind egyforma | <3,3',3''> | 3 | <3'',3'> |
| több egymás utáni remMax(), majd add hatása | <2,3,1> | 3 2 1 3 2 1 | <2,1> remMax() <1> remMax() <> add(3) add(2) add(1) remMax() <1,2> remMax() <1> remMax() <> |