Inf Progtech Übung: Priority Queue (Heap)

Klaus Kusche

Schreib ein Programm, das eine ***Priority Queue*** (einen ***Heap***) implementiert und testet (such dir die Details der Priority-Queue-Operationen wenn nötig am Internet).
Bei unserem Heap soll immer ***das kleinste Element vorne*** stehen.

- Definiere zuerst einen **`struct`**-*Typ* für eine Priority Queue.
  Er enthält einen **`int`** für die *Anzahl* der aktuell in der Queue befindlichen Werte und ein *Array für die Nutzdaten*.

  Als Nutzdaten verwenden wir normale **`int`**'s,
  die *Array-Größe* wird mittels **`define`**-*Konstante* festgelegt.

  Wie bei Priority Queues üblich bleibt das *Array-Element Nummer 0 unbenutzt*,
  die tatsächliche Kapazität der Queue ist daher *um 1 kleiner* als die Konstante.

- Die Funktion **`put`** wird mit einem *Pointer auf so eine Struktur*
  und einem **`int`** aufgerufen. Sie soll den **`int`** richtig in die Queue *einfügen*.

  `put` soll bei *Erfolg* **`true`** liefern und **`false`**, wenn die Queue schon *voll* ist.

  Die Grundidee ist, das neue Element *hinten* an das Array *anzuhängen*
  und so lange *mit seinem Vater zu vertauschen*, bis es am richtigen Platz steht
  (d.h. entweder *größer* als sein Vater ist oder *ganz vorne* gelandet ist).

- Die Funktion **`get`** wird mit einem *Pointer auf eine Queue-Struktur*
  und einem *"by Reference"* übergebenen **`int`**-Parameter aufgerufen.
  Sie soll den *kleinsten Wert* aus der Queue *entnehmen*
  und im **`int`**-Parameter *speichern*.

  `get` soll bei *Erfolg* **`true`** liefern und **`false`**, wenn die Queue *leer* ist.

  `get` ist komplizierter:

  - Der als *Ergebnis* zu liefernde Wert ist der *vorderste Wert im Array*.
  - Die *Lücke*, die er hinterlässt, wird mit dem *hintersten Wert aus dem Array* gefüllt.
    Das Array wird dadurch um ein Element *kürzer*.
  - Damit man wieder eine gültige Queue erhält, muss dieser Wert dann so lange
    mit dem kleineren seiner beiden Söhne vertauscht werden, bis beide Söhne *größer* als er selbst sind (oder er keine Söhne mehr hat).
    Achte dabei auch auf den *Sonderfall*, dass der Wert nur einen *einzigen* Sohn (am Array-Ende) hat!

- Das **Hauptprogramm** wird mit *beliebig vielen Zahlen auf der Befehlszeile* aufgerufen.

  Es legt eine Priority Queue an (was musst du bei einer neuen Queue *initialisieren*?)
  und *speichert alle Zahlen* der Befehlszeile mit **`put`** in dieser Queue (auf Überfüllung prüfen!).

  Dann ruft es mittels Schleife *immer wieder* **`get`** auf und *gibt* die gelieferte Zahl *aus*, bis die Queue *leer* ist.
  Wenn alles passt, sollte das alle Zahlen der Eingabe *aufsteigend geordnet* liefern.