

L4: Heap

Bekommen in L4

Abgabe in L5

Implementiere in C++ eine Anwendung für die gegebene Aufgabe und benutze dabei einen Heap als Datenstruktur. Falls es nicht explizit angegeben wird, dann benutze einen binären Heap. Für die Implementierung des Heaps dürft ihr kein STL vector oder andere Containers oder Datenstrukturen aus STL (oder aus andere Bibliotheken) benutzen.

1. Gegeben seien k sortierte Listen. Man muss die k Listen mergen (interclasare). Für die Implementierung der Input Liste benutze die list Klasse aus der STL Bibliothek.
2. Gegeben seien k sortierte Arrays. Man muss die k Arrays mergen (interclasare). Für die Implementierung des Input Arrays benutze die vector Klasse aus der STL Bibliothek.
3. Implementiere ADT Prioritätsschlange (Priority Queue) mithilfe eines ternären Heap (anstatt zwei Kinder, gibt es drei Kinder) als Repräsentierung.
4. Implementiere ADT Prioritätsschlange (Priority Queue) mithilfe eines 4-ären Heap (anstatt zwei Kinder, gibt es vier Kinder) als Repräsentierung.
5. Implementiere, mithilfe eines binären Heap, ein Container PQ2 ähnlich mit Priority Queue, mit der Unterschied, dass man das Element mit der zweithöchsten Priorität zurückgeben (top)/ löschen (pop) will (bezüglich einer Relation R zwischen den Elementen). Benutze einen binären Heap als Repräsentierung.
6. Implementiere, mithilfe eines binären Heap, ein Container PQ2 ähnlich mit Priority Queue, mit der Unterschied, dass man das Element mit der zweithöchsten Priorität zurückgeben (top)/ löschen (pop) will (bezüglich einer Relation R zwischen den Elementen). Benutze einen ternären Heap (anstatt zwei Kinder, gibt es drei Kinder) als Repräsentierung.
7. Implementiere, mithilfe eines binären Heap, ein Container PQ2 ähnlich mit Priority Queue, mit der Unterschied, dass man das Element mit der zweithöchsten Priorität zurückgeben (top)/ löschen (pop) will (bezüglich einer Relation R zwischen den Elementen). Benutze einen 4-ären Heap (anstatt zwei Kinder, gibt es vier Kinder) als Repräsentierung.
8. Implementiere, mithilfe eines binären Heap, ein Container PQ3 ähnlich mit Priority Queue, mit der Unterschied, dass man das Element mit der dritthöchsten Priorität zurückgeben (top)/ löschen (pop) will (bezüglich einer Relation R zwischen den Elementen). Benutze einen binären Heap als Repräsentierung.
9. Implementiere, mithilfe eines binären Heap, ein Container PQ k ähnlich mit Priority Queue, mit der Unterschied, dass man das Element mit der k -höchsten Priorität zurückgeben (top)/ löschen (pop) will (bezüglich einer Relation R zwischen den Elementen). Benutze binären Heap als Repräsentierung. Hint: benutze zwei binäre Heaps, einen mit der festen Länge k .

10. Gegeben sei ein Array mit n natürlichen unterschiedlichen Zahlen. Berechne die Summe der größten k Zahlen aus dem Array mithilfe eines Algorithmus mit $O(n \cdot \log 2 k)$. Für die Implementierung des Input Arrays benutze die vector Klasse aus der STL Bibliothek.
11. Gegeben sei eine Liste mit n natürlichen unterschiedlichen Zahlen. Lösche die k kleinsten Zahlen aus der Liste mithilfe eines Algorithmus mit $O(n \cdot \log 2 k)$. Für die Implementierung der Input Liste benutze die list Klasse aus der STL Bibliothek.
12. Gegeben sei ein Array mit unterschiedlichen Elementen vom Typ TComp. Man muss die k größten Elemente aus dem Array bestimmen (bezüglich einer Relation R zwischen den Elementen). Benutze dabei einen ternären Heap (anstatt zwei Kinder, gibt es drei Kinder). Für die Implementierung des Input Arrays benutze die vector Klasse aus der STL Bibliothek. Man darf den Input Array nicht sortieren. Falls $R \leq$ ist, dann ist das erste Element das Minimum.
13. Gegeben sei ein Array mit unterschiedlichen Elementen vom Typ TComp. Man muss die k größten Elemente aus dem Array löschen (bezüglich einer Relation R zwischen den Elementen). Benutze dabei einen 4-ärer Heap (anstatt zwei Kinder, gibt es vier Kinder). Für die Implementierung des Input Arrays benutze die vector Klasse aus der STL Bibliothek. Man darf den Input Array nicht sortieren. Falls $R \leq$ ist, dann ist das letzte Element das Maximum.
14. Gegeben sei ein Array mit unterschiedlichen Elementen vom Typ TComp. Man muss das Produkt der k größten Elemente aus dem Array bestimmen (bezüglich einer Relation R zwischen den Elementen). Benutze dabei einen n -ären Heap (anstatt zwei Kinder, gibt es n Kinder). Für die Implementierung des Input Arrays benutze die vector Klasse aus der STL Bibliothek. Man darf den Input Array nicht sortieren.