

Ejercicios de Repaso

Métodos de Selección - Montículos

Montículos (Binary heaps)

1. Proponer la implementación de las operaciones insertar, remover el máximo utilizando:

- Una cola
- Una pila

Para cada una de ellas indicar su tiempo (en número de comparaciones) y compararlo con la implementación en montículo.

2. Utilizando el resultado del punto 1, indicar porque no es eficiente mantener una variable auxiliar *máximo* que se actualice cada que se inserta un item. La variable sería muy efectiva para obtener el mayor valor.

3. Se tienen una colección de puntos (x,y,z) en el espacio. Dar un algoritmo para encontrar los 10 puntos más cercanos al origen.

4. En alguna aplicación se requiere una *cola de prioridad max/min*, es decir una cola que permita insertar elementos y remover eficientemente tanto el mayor como el menor elemento. Proponer una implementación eficiente de la *cola de prioridad max/min*.

5. (*) Se tiene una colección de puntos (x,y) en el plano. Se quieren encontrar los 10 pares de puntos más cercanos entre sí. Indicar el algoritmo y establecer su eficiencia.

Análisis de espacio

1. Cuánto espacio requiere el montículo de tamaño N representado en arreglo?
2. Se quiere experimentar con un montículo implementado por medio de clases auxiliares Nodo:

```
class Nodo {  
    Key key;  
    Nodo hijoIzq;
```

```
        Nodo hijoDer;  
    }
```

- a) Dar una implementación del montículo considerando sus operaciones básicas.
- b) Estimar el espacio requerido por esta implementación.

Algoritmos híbridos

Se tienen 3 métodos para resolver el problema de los Top-M. Proponer un algoritmo híbrido que en función del tamaño de la colección N y el número de elementos top M, utilice la solución más eficiente para una instancia dada del problema.