

Programación II

TPO. *Priority Queues y Heaps*

1. Filas de prioridad, montones y montículos

La implementación de filas de prioridad que hemos visto en clase fue realizada de manera ineficiente, pues dimos prioridad a la comprensión del tema sobre la eficiencia algorítmica. Recordemos que la inserción y la eliminación de un elemento sobre esta estructura tenían un costo de $\mathcal{O}(n)$ donde n es la cantidad de elementos presentes en la fila.

Existe una estructura de datos que no vimos en clase, pero que es importante. Son los *heaps* (que puede traducirse en español como “montones” o, a veces, “montículos.”) Esta estructura consiste, en esencia, de un árbol binario casi completo que puede representarse implícitamente con un arreglo, estático o dinámico. Se puede encontrar más información sobre *heaps* en la bibliografía propuesta.

El detalle del TPO se detalla a continuación.

1. Realice una investigación sobre la estructura de datos *heap*, en qué consiste y cómo puede ser implementada. Usted debe estar en condiciones de explicar todo esto en la exposición oral individual. Se incluyen en esto implementaciones estáticas y dinámicas y todos los cálculos de costos con sus correspondientes justificaciones.
2. Implemente en Java las referidas estructuras. En una primera etapa, puede implementar algunos ejemplos simples para verificar que todo funcione adecuadamente.
3. Implemente en Java los métodos de filas de prioridad utilizando esta estructura y efectúe los cálculos de costos correspondientes.
4. Implemente en Java algún otro ejemplo no trivial de su elección. Por ejemplo, *heapsort*.
5. Debe presentar un informe escrito con la siguiente estructura: una introducción, un breve desarrollo teórico, la explicación de las implementaciones, conclusiones y bibliografía utilizada (libros, artículos o sitios Web.) Cada ítem de la bibliografía debería estar citado al menos una vez en el texto. Las citas textuales deben obligatoriamente estar referenciadas.
6. El informe debe estar escrito en español o en inglés. La corrección de la escritura (en cualquiera de los idiomas) se evaluará también, como se hace en cualquier congreso.
7. Cada alumno recibirá una nota que reflejará la calidad del trabajo grupal y su propia exposición oral.

Importante: si bien el trabajo es grupal, la evaluación y la nota son individuales.

Algunos comentarios: este trabajo requiere una primera parte de investigación (muy básica, pero investigación al fin.) Esto significa que lo primero que debe hacerse es compenetrarse de las estructuras de datos que se utilizarán (los *heaps*) y asegurarse de que los principios teóricos y de funcionamiento se comprenden adecuadamente. Esto implica:

1. Buscar la información. Generalmente es necesaria más de una referencia para tener un panorama completo.
2. Evaluar la información encontrada (si se busca en Internet, no es lo mismo *rincondelvago* que un artículo de la Universidad de Princeton.) El criterio de selección de la bibliografía se evaluará.
3. Por supuesto, leer y comprender la información recogida.

Se espera que los grupos trabajen de manera autónoma, pero por supuesto pueden solicitar mi colaboración cuando la necesiten.

2. Fechas importantes

- 16 de junio: demostración de funcionamiento del TPO.
- 23 de junio: entrega final y defensa del TPO.

Referencias

- [1] Brassard, Gilles; Bratley, Paul: *Fundamentals of Algorithmics*, Prentice-Hall, 1996.
- [2] Cormen, Thomas; Leiserson, Charles; Rivest, Ronald: *Introduction à l'algorithme*, Dunod, 2004 (en francés.)
- [3] Cormen, Thomas; Leiserson, Charles; Rivest, Ronald: *Introduction to Algorithms*, MIT Press, 2002.
- [4] Goodrich, Michael; Tamassia, Roberto: *Algorithm Design and Applications*, John Wiley & Sons, 2015.
- [5] Kleinberg, Jon; Tardos, Éva: *Algorithm Design*, Pearson / Addison-Wesley, 2006.
- [6] Sedgewick, Robert; Wayne, Kevin: *Algorithms*, Addison-Wesley, 2011.
- [7] Skiena, Steven: *The Algorithm Design Manual*, Springer, 2012.