Proyecto 2

Análisis de Algoritmos
Primer Semestre 2024, Prof. Cecilia Hernández

Ayudantes: Martita Muñoz, Álvaro Guzmán

Fecha Inicio: Domingo 2 de junio 2024.

Fecha Entrega: Lunes 24 de junio 2024 (23:59 hrs).

Instrucciones de entrega:

- Se debe entregar (a través de Canvas) el informe con sus respuestas en formato pdf y un .zip con los archivos o programas que utilice para sus respuestas. Es decir, en su entrega de Canvas deberá aparecer un archivo pdf (informe) y un archivo zip con los códigos o material que referencie en su informe.
- El informe debe estar escrito en LaTeX.
- Sólo una persona del grupo debe enviar la entrega por Canvas, procurando que los integrantes del grupo estén detallados en su informe.
- No se aceptan grupos de una persona.
- El código fuente deberá ir en el .zip que entregue y deberá estar escrito en C/C++.

Pregunta 1 [1.2 puntos]: Considere un sistema de cloud computing que tiene disponibilidad de N unidades de procesamiento p_i donde cada una tiene una capacidad c_i para atender un requerimiento. Por otro lado, existen N usuarios, u_i donde cada uno de ellos solicita un requerimiento que requiere r_i capacidad de procesamiento. Asuma que el número de unidades de procesamiento es la misma que el número de usuarios, pero no puede preprocesar la información otorgada para las unidades de procesamiento y usuarios.

Ejemplo: Entradas: $N = 7$, arreglo P que contiene capacidades de procesamiento de cada unidad, arreglo U que contiene los requerimientos de cada usuario. Salida: arreglo A_U que indica para cada usuario el identificador de la unidad de procesamiento que lo debe atender.

$$P = [8, 20, 10, 3, 50, 2, 5]$$

$$U = [10, 2, 50, 8, 20, 5, 3]$$

Asignación de usuarios a unidades de procesamiento:

$$A_U = [2, 5, 4, 0, 1, 6, 3]$$

1. Escriba un algoritmo aleatorizado que resuelve el problema de asignación de requerimientos de usuario en unidades de procesamiento.
2. Proporcione el análisis de tiempo esperado.
3. Proporcione la implementación, evaluación y discusión.

Pregunta 2 [1.2 puntos]: Considere un grupo de n estudiantes universitarios. Se desea construir un arreglo que contenga los promedios de notas de los n estudiantes y que soporte las operaciones de inserción y de búsqueda, para ello se puede mantener un arreglo ordenado, de modo que la búsqueda tomará $O(\log n)$ y la inserción, $O(n)$.

Si bien la búsqueda es rápida, la inserción no lo es tanto. Se le pide diseñar una nueva estructura de modo que la inserción sea más rápida, para ello siga la siguiente estrategia:

i) Se definen los arreglos **ordenados** $A_0, A_1, ..., A_{k-1}$ de modo que A_i es de tamaño 2^i, con $i = 0, 1, ..., k-1$.

ii) Sea $\langle n_{k-1}, n_{k-2}, ..., n_0 \rangle$ la representación binaria de n. Es decir, $n = \sum_{i=0}^{k-1} 2^i n_i$.

iii) Los elementos (los n promedios) se particionan en los arreglos $A_0, A_1, ..., A_{k-1}$ de modo que A_i estará lleno si $n_i = 1$, y estará vacío si $n_i = 0$.

iv) Si bien cada arreglo A_i está ordenado, ellos no tienen ninguna relación entre si.

A modo de ejemplo, observe en la Figura 1 como se puede visualizar esta estructura de datos al agregar los elementos de una lista de promedios dada por [2.7, 4.9, 7, 5.56].

Se pide realizar lo siguiente:

1. Dado $n \in \mathbb{N}$, encontrar el menor valor de $k \in \mathbb{N}$ de modo que los n promedios se puedan particionar en los k arreglos siguiendo la estrategia anterior. Justifique. Utilice este valor de k para el resto del problema.
2. Muestre, justificadamente, que la suma de las cantidades de elementos en los k arreglos es igual a n.
3. Definir una operación Buscar para esta estructura de datos (los k arreglos) de modo que el tiempo de ejecución en el peor caso sea de $O(\log^2 n)$. Justifique.
4. Definir una operación Insertar para esta estructura de datos (los k arreglos) de modo que el tiempo de ejecución en el peor caso sea de $O(n)$, pero que el costo amortizado de esta sea de $O(\log n)$. Muestre la complejidad amortizada para M inserciones usando el método de contabilidad. (**Pista:** Puede suponer que unir dos arreglos ordenados de tamaño m toma un tiempo de $2m$). Notar que la operación Insertar debe insertar el nuevo elemento de modo que se mantenga la estructura, es decir, tener k arreglos $A_0, A_1, ..., A_{k-1}$ de modo que todos ellos estén ordenados y estén llenos o vacíos según el punto iii) descrito en la estrategia (puede ver la Figura 1 para guiarse).
5. Escriba un programa que genere listas de tamaño n de promedios (números reales entre 1.0 y 7.0) y utilícelo para validar sus resultados teóricos respecto a las operaciones Buscar y Insertar del item 3. y 4. de forma experimental. Utilice tamaños suficientemente grandes (n=1000, 2000, ...) y obtenga al menos 10 mediciones por experimento. Grafique y discuta sus resultados.

Pregunta 3 [1.2 puntos]: En clases usted estudió una forma de resolver el problema de Vertex-Cover aproximado, llamemos a esta estrategia A. Se propone la siguiente heurística para resolver el mismo problema. Iterativamente se va escogiendo el vértice v de mayor grado (si hay empate no importa el orden) y se eliminan todas las aristas que tengan a v como uno de sus extremos. Llamemos a esta segunda estrategia B.

Se pide lo siguiente:

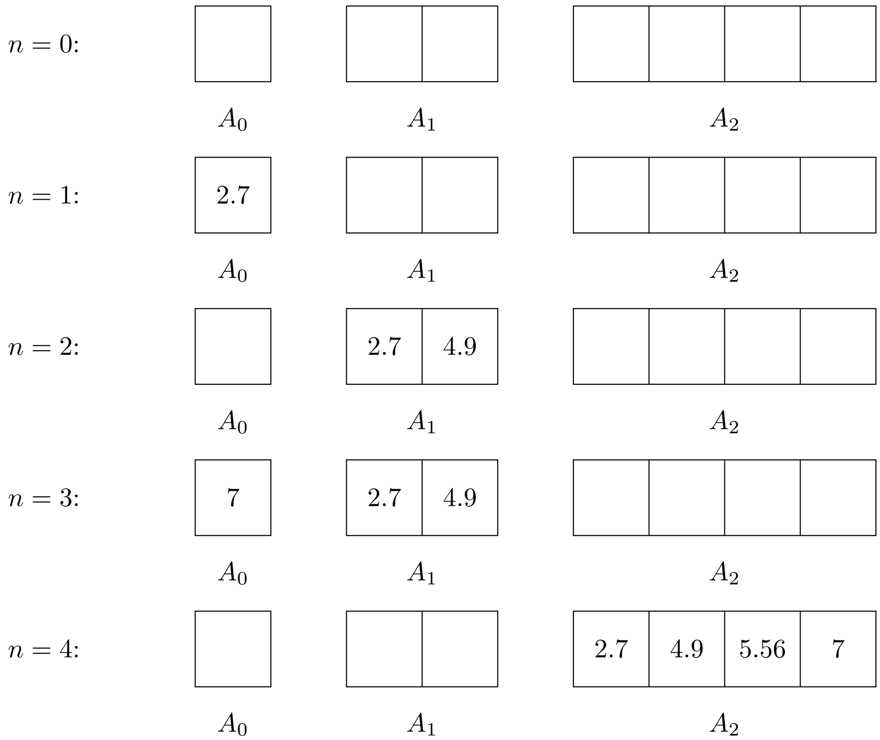

Figura 1: Nueva estructura de datos. Notar que los arreglos A_i vacíos y llenos se corresponden con la representación binaria de n en cada caso, según lo establecido en la parte iii) de la estrategia. Además, estos arreglos siempre están ordenados. Se utilizó $k = 2$ a modo de ilustrar la estructura, pero en general el valor de k depende de n.

1. Implemente en C/C++ ambas estrategias A y B.
2. Evalúe su implementación en el siguiente grafo: *Enlace de descarga del grafo*. Encuentre la solución exacta y la solución entregada por ambas estrategias. ¿Cuál estrategia entregó una mejor solución? ¿Se cumple que la estrategia A es 2-aproximación? Justifique.
3. Evalúe su implementación con el siguiente grafo: *Enlace de descarga del grafo*. Discuta cuál estrategia dio mejores resultados. Según lo visto en clases, ¿qué puede decir de la solución real? (No se pide encontrar la solución real)
4. Considere el grafo bipartito presentado en la Figura 2. Calcule la solución exacta y determine el razón de aproximación dado por ambas estrategias, en el caso de la estrategia B, comience por el vértice 6. ¿Qué puede concluir respecto al razón de aproximación para la estrategia B?

Observación: En los archivos *grafo1.txt* y *grafo2.txt* correspondientes a los grafos encontrará información de ellos en la primera línea para que así los pueda leer.

Pregunta 4 [1.2 puntos]: Una escuela necesita llevar a sus estudiantes, maestros, apoderados e invitados a las Olimpiadas a realizarse en otra ciudad. Para ello, dispone de m autobuses, los cuales tienen P asientos. Las personas que irán forman parte de uno de los n equipos que participarán en las Olimpiadas. Tenga en cuenta que el director desea que cada equipo completo vaya al interior de un autobús, pero puede darse el caso que un autobús se complete, por lo que algunas personas tendrán que ir en otro autobús alejado de su equipo. Su trabajo es asignar los equipos en los autobuses de forma tal que la cantidad de personas alejadas

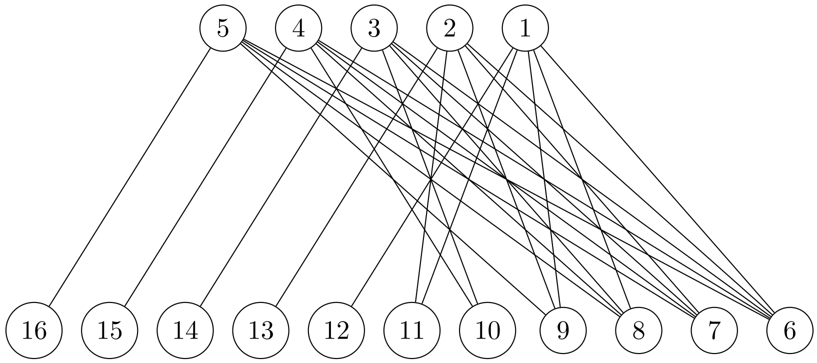

Figura 2: Grafo bipartito. Los grados de los vértices 1, 2, 3, 4 y 5 son iguales a 5. Los grados de los vértices 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 y 16 son 5, 4, 4, 3, 2, 2, 1, 1, 1, 1 y 1, respectivamente.

de sus equipos sea mínima. De ser posible, consiga que todos los equipos se encuentren completos al interior de un bus.

Sea w_i la cantidad de miembros pertenecientes al equipo i ($1 \leq i \leq n$), conformados por estudiantes, maestros, apoderados e invitados. Asuma que $\Sigma_{i=1}^{n} w_i \leq mP$, es decir, existen suficientes asientos para todas las personas. Además, asuma que $w_i \leq P$, es decir, que un equipo completo puede caber dentro de un autobús. También asuma que $1 \leq n \leq 2m$.

Ejemplos:

Suponga el siguiente caso: Tiene 4 autobuses con 50 asientos para 6 equipos ($m = 4$, $n = 6$, $P = 50$). La cantidad de miembros de cada equipo son los siguientes: $\{40, 20, 40, 50, 10, 30\}$. En este caso, en el primer autobús iría el equipo de 50, en el segundo iría un equipo de 40, en el tercer autobús iría el otro equipo de 40 y el equipo de 10, y en el cuarto autobús iría el equipo de 20 y el equipo de 30. No existen personas alejadas de sus equipos.

Suponga el siguiente ejemplo: Tiene 4 autobuses con 50 asientos para 6 equipos ($m = 4$, $n = 6$, $P = 50$). La cantidad de miembros de cada equipo son los siguientes: $\{40, 30, 40, 50, 10, 30\}$. En este caso, en el primer autobús iría el equipo de 50, en el segundo iría un equipo de 40, en el tercer autobús iría el otro equipo de 40 y el equipo de 10, y en el cuarto autobús iría un equipó de 30 y 20 miembros del segundo equipo de 30. Quedarían 10 personas alejadas de sus equipos.

Pregunta 5 [1.2 puntos] Una fábrica de cajas fabrica cajas de todos los materiales y tamaños posibles. La empresa busca guardar sus cajas ocupando el menor espacio posible, guardando las cajas dentro de otras cajas.

Dada n cajas, cada caja c_i tiene dimensiones x_i, y_i y z_i y un peso w_i. Para evitar dañas las cajas, una caja más pesada no puede ser guardada en una caja más ligera. Su trabajo consiste en encontrar la mayor cantidad de cajas que pueden ser guardadas dentro de otras cajas.

Ejemplos:

Suponga el siguiente ejemplo: Tiene 3 cajas disponibles ($n = 3$) con las características (x_i, y_i, z_i, w_i): $\{(40, 50, 30, 3), (30, 25, 40, 1), (41, 49, 30, 2)\}$. Es posible guardar las tres cajas metiendo c_3 dentro de c_1 y c_2 dentro de c_2. La mayor cantidad de cajas que pueden ser guardadas dentro de otras cajas es 3.

Suponga el siguiente ejemplo: Tiene 4 cajas disponibles ($n = 4$) con las características (x_i, y_i, z_i, w_i): $\{(40, 50, 30, 2), (21, 22, 23, 1), (30, 51, 40, 1), (41, 49, 30, 3)\}$. Es posible guardar c_2 en cualquiera de las otras tres cajas. La mayor cantidad de cajas que pueden ser guardadas dentro de otras cajas es 2.