

ALGORITMOS Y COMPLEJIDAD – MODALIDAD LIBRE

1er ETAPA 18/8/15

- 1- Verdadero o falso, justificar
 - a. $n^{\log_b a}$ pertenece a $O(n^{\log_c a})$ para todo a, b, c
 - b. $\Omega = (f(n) + g(n)) = \Omega = \max(f(n), g(n))$
 - c. Es falso que existe un algoritmo con tiempo de ejecución $\theta(n^2)$ y que el peor caso sea $O(n^3)$
- 2- Resolver las recurrencias con técnicas vistas en clases, indicar si no se puede resolver con alguna.
 - a. $T(n) = 3T(n/3) + \sqrt{n}$
 - b. $T(n) = \frac{1}{2}T(n/2) + n$
- 3- Dar contra ejemplo del problema de viajante con greedy
- 4- Dar algoritmo y tiempo de ejecución de búsqueda binaria dividiendo el algoritmo en 1/3 y 2/3
- 5- Dar algoritmo de puntos de articulación con un solo recorrido, indicar tiempo de ejecución
- 6- Aplicar Edmons Karp a una red de flujo dada describiendo cada paso detalladamente, indicar cuando se llega al máximo flujo
- 7- Definir skew heaps, análisis amortizado de sus operaciones
- 8- Demostrar

Lema 8

Sean $\mathcal{C}_1 \subset \mathcal{C}_2$ dos clases de complejidad tal que \mathcal{C}_1 es cerrada c.r. a \leq_r . Entonces cualquier $L \in \mathcal{C}_2$ que sea \mathcal{C}_2 -completo c.r. a \leq_r es tal que $L \notin \mathcal{C}_1$.

2da ETAPA 20/8/15

Para el problema del viajante

- Dar algoritmo de programación dinámica, justificar estrategia
- Análisis tiempo y espacio
- Implementación
- Análisis empírico: tamaño del grafo, densidad del grafo
- Conclusiones

3ra ETAPA 24/8/15

- 1- Dar algoritmo Prim, tiempo de ejecución de distintas implementaciones, demostrar su correctitud
- 2- Era un ejercicio de un arreglo de componente ordenados que tenía un corrimiento K:
Sea $A = 35-42-5-15-27-29 \rightarrow k=2$, $A = 27-29-35-42-5-15 \rightarrow k=4$
 - a. dar un algoritmo que dado un k retorne el elemento máximo en $\theta(1)$
 - b. dar un algoritmo DyC tal que no se conoce el k , retornar el máximo elemento

- 3- dar recurrencia, tiempo de ejecución y espacio del problema de triangularización de polígonos de programación dinámica
- 4- dar algoritmo para indicar si existe o no orden topológico de un grafo; tiempo de ejecución y demostrar su correctitud
- 5- Fibonacci heaps definir, sus operaciones, análisis de operaciones en el mejor caso, análisis amortizado de sus operaciones. Comparación con colas binomiales.
 - a. Enuncia y demuestre por que tales heaps llevan el nombre Fibonacci
- 6- Definir algoritmo probabilístico sesgado. Por qué es útil esta propiedad. Ejemplo de algoritmo sesgado y no sesgado.
- 7- Definir NPC
 - a. Teorema Cook, importancia, esquematizar demostración
 - b. Reducción Camino Hamiltoniano \leq_p Problema del viajante, demostrar que el problema del viajante pertenece a NPC.