In a recursion tree, each node represents the cost of a single ________ somewhere in the set of recursive function invocations. *

- ○ Problem
- ◉ sub problem
- ○ instruction
- ○ node
- ○ Other: ____________________

Solve the following recurrence using Master's theorem. *

$$T(n) = T(n/2) + 2^n$$

- [ ] T(n) = O(n*n)
- [ ] T(n) = O(n*n log n)
- [ ] T(n) = O(2^n)
- [x] cannot be solved

Recursion trees are particularly useful when the recurrence describes the running time of a ________ algorithm. *

- ○ Dynamic
- ○ Greedy
- ◉ Divide-and-conquer
- ○ Backtracking

Solve $T(n) = 8T(n/2) + n^2$ *

- ○ $O(n^2)$
- ○ $O(n)$
- ◉ $O(n^3)$
- ○ $O(n^2 \log n)$

Solve the recurrence by master method

$$T(n) = T(\sqrt{n}) + 1$$

(a) $T(n) = \Theta(loglogn)$

(b) $T(n) = \Theta(logn)$

(c) $T(n) = \Theta(\sqrt{n})$

(d) None of these.

- (●) a
- ( ) b
- ( ) c
- ( ) d

Under what case of Masters theorem will the recurrence relation of merge sort fall?

(a) 1^{st} case

(b) 2^{nd} case

(c) 3^{rd} case

(d) It cannot be solved using masters theorem

- [ ] a
- [x] b
- [ ] c
- [ ] d

If $T(n) = T(n/4) + T(n/2) + n^2$,then using recursion tree method

(a) $T(n) = \theta(n)$.

(b) $T(n) = \theta(n^2)$.

(c) $T(N) = \theta(n^3)$.

(d) None of the above

- ○ a
- ◉ b
- ○ c
- ○ d

Recurrence relation for binary search

(a) $T(n) = 2T(n/2) + \theta(1)$

(b) $T(n) = T(n/2) + \theta(1)$

(c) $T(n) = 2T(n/2) + \theta(n)$

(d) $T(n) = 2T(n/2) + \theta(n^2)$

Assume that a merge sort algorithm in the worst case takes 30 seconds for an input of size 64. Which of the following most closely approximates the maximum input size of a problem that can be solved in 6 minutes?

(a) 256

(b) 512

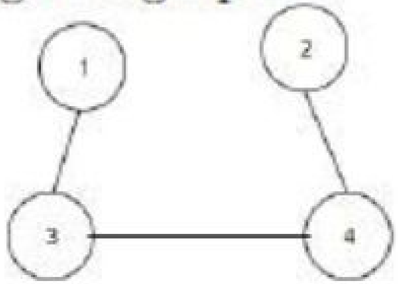
(c) 1024

(d) 2048

- ○ a
- ◉ b
- ○ c
- ○ d

*

What would be the number of zeros in the adjacency matrix of the given graph?



(a) 10
(b) 6
(c) 16
(d) 0

- [x] a
- [ ] b
- [ ] c
- [ ] d