②

QUIZ ONE

1) a) True. b/c $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$

b) True if $f(n) = O(h(n))$ $h(n) = \Omega(f(n))$

c) False. $O(\ )$ only indicates that the two are assymptotically equivalent, they may have different constant

d) True. They are asymptotically equivalent

e) True: ~~...~~ The list goes through n times, each going another $\frac{1}{2}n$ times (amorti... with splicing being a con... time operation. X

X

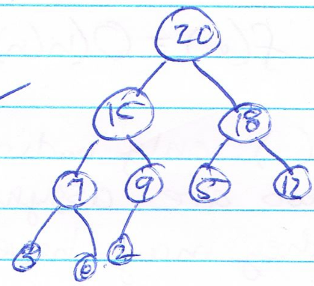
False, slicing takes linear time.

2) First, create a hash table to store unmatched elements. Next step through the two linked lists, in alternating sequence ~~when you~~ For each number, check if it is in the hash table. If it is, delete the number and subtract the counter by one. If it isn't add it to the table and increase the counter by one. When you reach the end of ~~one~~ a list, if the other still has elements (unless it only has one element left and is the 2^{nd} LL) return false. Otherwise, both ~~elements~~ s... have the same size. Check now that the counter = 0 ~~...~~ and return true if it does.

Hilroy

3) $p = \left(1 - \frac{1}{m}\right)^n$

4) a) True:



b) The smallest element must reside in a leaf ~~node~~, or $i \geq \lceil \frac{n}{2} \rceil$, for slot i in n nodes.

c) i) Each insert operation runs in $O(\lg n)$ time, or the height of the heap tree, since it involves swapping up $\lg n$ times in the worst-case. With n insertions, the running time is $n \lg n$

ii) In the worst case, each element takes $\lg n$ ~~times~~ swaps to insert, which occurs n times.

5) a) Splicing a list, $O(n)$

b) False, the keys must be known in advance.

c)

$$T(n) = \begin{cases} O(1), & \text{if } n=1 \\ T(n-1)+T(n-1), & \text{else} \end{cases}$$

Substitution Method, guess $T(n) = O(2^n)$

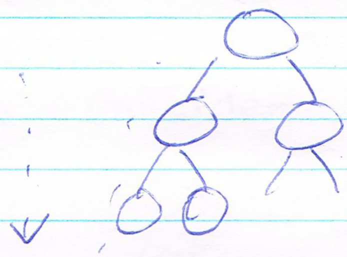
$T(n) < 2^n$

Base Case: $O(1) < 2^1$ TRUE

Recursive Case:

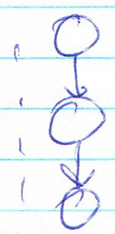
$T(n-1) + T(n-1) < 2^n$

RECURRENCE TREE



n levels

4) $T(n) = T(n-1) + O(1)$



n levels

$= O(n)$ ✓ $= 2+3+4+5+6+ \cdots +n$

5) $T(n) = T(n-1) + \lg T(n-1)$

$T(n) = \Theta(n \lg n)$ ✗

~~Each operation takes~~

6) a) False, worst-case is $O(n)$ if all keys hash to the same slot.

b) False, the probabilities of a slot being filled are no longer independent.

c) m is not prime, therefore many keys will hash to the same ~~slot~~ slot if they are a factor of m.

QUIZ #2:

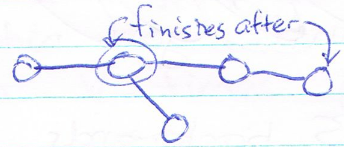
$\boxed{1}$

1) False, comparison sorts require $n \lg n$ in the worst case which

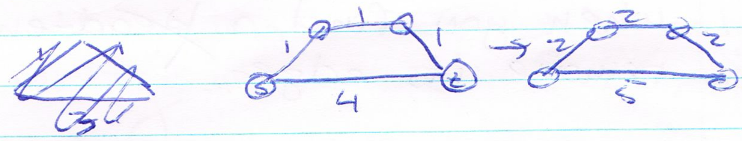
$$5 \lg 5 > 6$$

2) False. Heapsort is not a stable sort.

3) False.



4) False. Consider



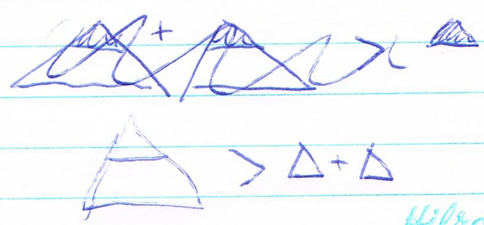
5) False, only with a stable sort

6) True. X «False, non-negative edges»

7) True.

$\boxed{2}$

1) the size of the first $\lg n / 2$ levels of the BFS tree was $\ll$ than the second $\lg n / 2$ levels



2) ~~O(lg V)~~ O(V+E) ~~Djikstra's~~

BFS.

3

~~First search for the first degree zero node~~

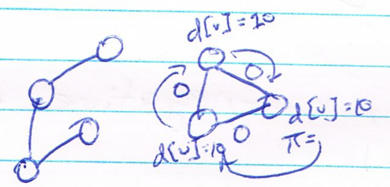
~~Pick a node s, trace backwards along its incoming edges until you.~~

Do a BFS backwards from any starting node s, onto its incoming edges, ~~then~~ when you find a ~~dead~~-end, pop it and remove its edges.

If G has cycles this returns an infinite loop.

4

1)



2) ~~Check that~~ ~~π~~

Backtrack through all $\pi[\ ]$ to ensure that they never repeat.

3) ~~Run Bellman Ford on each node and return the maximum value found, then~~

Run Bellman Ford and return the longest shortest path, then repeat this for each node, returning the maximum of that

BF has running time of $O(VE)$
Therefore this algorithm is $O(V^2E)$ since it runs BF V times ($O(n^3)$ if $n = V+E$)

4) Beginning at s, run a BFS which only considers vertexes which are $\delta(s, u) \leq \delta(s, t)$

Begin at t!