CS300 Introduction to Algorithm (Sample Problems for Midterm)

1. Suppose that you have a collection of n nuts and n bolts, where each bolt fits into exactly one nut. Visually, you cannot find any difference, only by testing a bolt with a nut you can determine whether the bolt is too thick, too thin, or it fits. Design an efficient algorithm to assign each bolt to the corresponding nut and analyze your algorithm.

Ref: 1st Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

Edited by 20170504 Juan Lee

2. (a) Describe a simple algorithm to find the minimum and the maximum of a sequence of n elements (of a linearly ordered universe) and analyze the exact number of comparisons needed as a function of n.

   (b) Solve the same problem with divide-and-conquer by partitioning sequences of length greater than 2 into two halves. Give an exact recursive equation for the number $C(n)$ of comparisons. You may assume that n is a power of 2.

3. Consider the following Python function to compute the n-th Fibonacci number f_n:

```
def fib(n):
    if n==0: return 0
    elif n == 1: return 1
    else: return fib(n-1) + fib(n-2)
```

(a) Show that this function has a runtime exponential in n. *Hint:* Count just the number $T(n)$ of additions. Set up a recursive equation for $T(n)$ and show by induction that it equals $f(n+1) - 1$ for all n.

(b) What is, as a function of n in terms of θ, the space needed by this algorithm?

(c) Find a function of linear runtime and constant space for the same problem.

(d) Find a function of runtime $\theta(\log n)$ for the same problem.

Hint: Observe that for $n \geq 2$,

$$\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = A \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix} \text{ where A is the matrix } A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Ref: 2nd Homework Assignment of CS300, prof. Helmut Alt, 2017 Spring

Edited by 20170504 Juan Lee

4. (a) Show that an algorithm to find the median in linear time can be used to solve the general selection problem in linear time.

   (b) Suppose you are given a deterministic (i.e. not randomized) algorithm A of linear runtime for the selection problem (such algorithms exist). Explain how A can be used to design a deterministic variant of Quicksort of runtime $O(n \log n)$ in the worst case.

5. (a) Give the comparison tree for Mergesort (splitting into subsequences of sizes $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$) and $\mathrm{n} = 3$.

   (b) Show by a comparison tree argument that any comparison based algorithm for searching an array $\mathrm{A}[1], \mathrm{A}[2], \dots, \mathrm{A}[\mathrm{n}]$ needs at least $\lceil \log n \rceil$ comparisons in the worst case.

6. (a) Given an adjacency-list representation of a directed graph, how long does it take to compute the outdegree of every vertex? How long does it take to compute the indegrees?

   (b) The transpose of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$ where $E^T = \{(v, u) \in V \times V | (u, v) \in E\}$. Describe and analyze efficient algorithms for computing G^T from G for both the adjacency-list and the adjacency-matrix representations of G.

7. Determine and analyze efficient algorithms for the following graph problems.

   (a) Is an undirected graph acyclic, i.e., a forest? *Hint:* dfs.

   (b) Is a directed graph acyclic?

   (c) Find a path which traverses each edge of an undirected connected graph at least once. *Hint:* dfs.

   (d) Find a cycle in an undirected connected graph which traverses each edge exactly once, if such a cycle exists. *Hint:* Show that this is the case, exactly if all vertices have an even degree.

8. (a) For directed and undirected graphs give the different types of edges (forward-, backward-, tree-, and cross-edges) a graph can have based on a dfs-tree. Explain your answer.

   (b) Solve the same problem for bfs-trees.

9. Show that any directed acyclic graph (dag) has at least one vertex with outdegree 0 and at least one with indegree 0.

10. (a) Modify Dijkstra's algorithm so that it not only gives the lengths of the shortest paths but also a structure for finding the shortest paths themselves.

    (b) What is the space requirement of the algorithm of Floyd-Warshall for a graph with n vertices if the values $\mathrm{d}_{i,j}^{\mathrm{k}}, k = 0, \dots n; i, j = 1, \dots, n$ are just stored in a three-dimensional array? How can it be improved?

    (c) Modify Floyd-Warshall's algorithm so that it not only gives the lengths of the shortest paths but also a structure for finding the shortest paths themselves. *Hint:* Compute each predecessor vertex $\pi_{i,j}^{\mathrm{k}}$ of vertex j in some shortest path from i to j with intermediate vertices in $\{1, \dots, \mathrm{k}\}$.

11. Describe an efficient divide-and-conquer algorithm to find both maximum and minimum values of n elements in an array. Analyze the number of comparisons of your algorithm.

12. The running time of QUICKSORT depends on both the data being sorted and the partition rule used to select the pivot. Suppose we always pick the pivot element to be the key of the median element of the first three keys of the subarray. On a sorted array, determine whether QUICKSORT now takes $\theta(n), \theta(n \log n), \theta(n^2)$. Explain why.

13. Suppose an array A consists of n elements, each of which is red, white, or blue. We seek to rearrange the elements so that all the reds come before all the whites, which come before all the blues. The only operation permitted on the keys are:

    EXAMINE(A, i) – report the color of the i-th element of A

    SWAP(A, i, j) – swap the i-th element of A with the j-th element.

    Give a linear time procedure.

Ref: CS300 Midterm Exam, 2005 Spring

Edited by 20170504 Juan Lee

14. Assume that in a binary search tree, it costs α to go left and β to go right. Assuming that all searches are successful, describe an algorithm to construct an optimal binary search tree given keys $k_1, \ldots, k_n (k_1 < \cdots < k_n)$ with probabilities of being searched $p_1, \ldots, p_n$.

Ref: CS300 Midterm Exam, Spring 2005

Edited by 20170504 Juan Lee

15. Describe an $O(n \log k)$ algorithm which merges k sorted lists with the total of n elements into one sorted list. (Use heap) Explain briefly why it takes $O(n \log k)$.

16. Consider a set of n intervals $[a_i, b_i]$. We say a set S of intervals covers the unit interval [0,1] if the union of the intervals S contains the unit interval [0,1].

    (a) Given a set of n intervals that covers [0,1], describe an algorithm that computes a minimum subset of the intervals that also covers [0,1].

    (b) Analyze the running time of your algorithm (in a few sentences)

    (c) Prove the correctness of your algorithm.

Ref: CS300 Midterm Exam, 2005 Spring

Edited by 20170504 Juan Lee

17. Give True or False for each of the following statements. Justify your answers.
    (1) We can sort n integers in the range 0 to $n^3 - 1$ in O(n) time.
    (2) $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$
    (3) $f(n) = \theta(f\left(\frac{n}{2}\right))$
    (4) If we use a d-ary heap instead of binary heap, EXTRACT-MAX operation that extracts the maximum element from the heap takes $\theta(\frac{\log n}{\log d})$
    (5) We can find 2 smallest of n element with $n + \lceil \log n \rceil - 2$ comparisons.
    (6) If we use groups of 7, instead of 5, in the worst-case linear-time selection algorithm, the algorithm still works in linear time. Justify your answer by giving a recurrence for this algorithm.

Ref: CS300 Midterm Sample Problems, prof. 최성희

Edited by 20170504 Juan Lee

18. Consider the following divide-and-conquer algorithm to multiply two n-bit numbers X and Y. We divide X into A and B and B and Y into C and D, where A, B, C, D are n/2-bit numbers.

$$\mathrm{X} = 2^{\frac{\mathrm{n}}{2}}\mathrm{A} + \mathrm{B}$$

$$\mathrm{Y} = 2^{\frac{\mathrm{n}}{2}}C + D$$

$$\mathrm{XY} = 2^{\mathrm{n}}AC + 2^{\frac{n}{2}}BC + 2^{\frac{n}{2}}AD + BD$$

(a) Give a recurrence for the running time of the algorithm and give a tight asymptotic bound.

(b) Observe that we can also obtain XY as follows:

$$\mathrm{XY} = \left(2^{\mathrm{n}} - 2^{\frac{n}{2}}\right)AC + 2^{\frac{n}{2}}(A+B)(C+D) + \left(1 - 2^{\frac{n}{2}}\right)BD$$

Give a recurrence for the running time of this algorithm and give a tight asymptotic bound.

19. Solve the following recurrences by giving tight θ-notation bound. Justify your answer.

(1) $\mathrm{T(n)} = 7\mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + n^2$

(2) $\mathrm{T(n)} = 2\mathrm{T}\left(\frac{\mathrm{n}}{3}\right) + \mathrm{n}\log n$

(3) $\mathrm{T(n)} = 4\mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + n^3$

(4) $\mathrm{T(n)} = \mathrm{T}\left(\frac{\mathrm{n}}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + n$

20. Give True or False for each of the following statements. Justify your answers.
    (1) $f_1(n) = \theta(g_1(n))$ *and* $f_2(n) = \theta(g_2(n))$. Then, $f_1(n)f_2(n) = \Omega(g_1(n)g_2(n))$
    (2) There exists a comparison sort of 6 numbers that uses at most 9 comparisons in the worst case.
    (3) We can use QUICKOSORT as the intermediate sort of radix sort.
    (4) The running time of QUICKSORT when all elements of array A have the same value is $O(n \log n)$

Ref: CS300 Midterm Sample Problems, prof. 최성희

Edited by 20170504 Juan Lee

21. We are given n points in the unit circle, $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for $i = 1,2,\ldots,n$. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. Design an algorithm with an expected running time of $\Theta(n)$ to sort the n points by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin. Explain briefly and analyze the running time.

22. Let $X[1 \dots n]$ and $Y[1 \dots n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\log n)$-time algorithm to find the k-th smallest element of all 2n element in array X and Y. (You may assume that all elements are distinct).

23. We are looking at the price of a given stock over n consecutive days, $i = 1,2, \ldots, n$. For each day I, we have a price $p(i)$ for the stock on that day. We'd like to know: How should we choose a day I on which to buy the stock and a later day $j > i$ on which to sell it, if we want to maximize the profit, $p(j) - p(i)$? (profit can be negative, if there is no way to make money during the n days.)

    (1) Using Divide-and-Conquer approach, find the optimal profit, such that p(j)-p(i) is maximized. Explain your algorithm and analyze the running time.

    (2) Using Dynamic Programming, find the optimal profit in $O(n)$ time. Explain your algorithm briefly and analyze the running time and space requirements.

24. Suggest four sorting algorithms, Bogo Sort, Selection Sort, Merge Sort and Quick Sort, and analyze the number of comparisons of each sorting algorithms.

25. (a) Describe Turing Machine and state the Time Complexity and Space Complexity.

    (b) Describe Random Access Model(RAM) and state two criterions for analyzing.

    (c) State the Church-Turing Thesis and Theorem of relationship between the Time Complexity of logarithmic cost criterion of RAM and runtime of Turing Machine.

26. (a) Define O, Θ, o, Ω notations.

    (b) Determine where $n!$ fits it.

Ref: CS300, 2017 Spring

Edited by 20170504 Juan Lee

27. Describe QUICKSELECT and analyze the Time Complexity.

28. Prove that any comparison-bases algorithm has a runtime of $\Omega(\mathrm{n} \log n)$.

29. (a) State and analyze Binary Search.

    (b) State and analyze Interpolation Search.

30. State two data structures for representing the Graph, and give some weaknesses of each structures.

31. (a) Describe Depth-First-Search algorithm and give the runtime of DFS.

    (b) Prove that dfs(s, G) visits exactly all vertices in G to which a path from s exits.

    (c) Describe the edges for directed and undirected dfs-tree.

32. (a) Describe Breadth-First-Search algorithm and give the runtime of BFS.

    (b) Prove that the path that is found (tree-edges in bfs-tree) from s to v is the shortest path from s to v.

    (c) Describe the edges for directed and undirected bfs-tree.

33. (a) Describe Topological Sorting and give the runtime of Topological Sorting.

    (b) Prove that every directed acyclic graph(dag) contains at least one vertex of indegree 0.

34. (a) Describe Dijkstra's Algorithm and give runtime of Dijkstra's Algorithm.

(b) Prove that when $u \in V$ gets inserted into S(initially $\emptyset$ in Dijkstra's Algorithm), D[u] is the length of the shortest path from s to u.

(c) What is the runtime for determining all pairs of shortest path using Dijkstra's Algorithm?

35. (a) Describe Floyd-Warshall Algorithm and give the runtime of Floyd-Warshall.

    (b) Give at least three other uses of Floyd-Warshall with small change in Algorithm.

    (c) Explain Dynamic Programming.

36. (a) Define the minimum cost spanning tree(MST).

    (b) Describe Prim's Algorithm and give the runtime of Prim's Algorithm.

    (c) Prove that Prim's Algorithms really construct MST. (correctness)

Ref: CS300, 2017 Spring

Edited by 20170504 Juan Lee