亞洲大學 博士班資格考 試題

文件編號：AC-0054
機密等級：2 限閱

考試日期： 111 年 8 月 3 日

| 考試科目 | 演算法 | 系所別 | 人工智慧博士學位學程 | 命題教師 | |
|---|---|---|---|---|---|

1. (10%) Prove or disprove that the binary search is optimal over all searching algorithms that perform comparisons only.

2. (20%) An ordinary version of the Quick-Sort algorithm picks up a random element, called "pivot", from the input sequence $S[1\ldots n]$ of numbers to partition $S[1\ldots n]$. To improve the Quick-Sort algorithm, a potential direction is to choose a "better" pivot as follows: Choose as the pivot $x$ the median of a set of three numbers randomly selected from $S[1\ldots n]$. For the ease of analysis, assume that the numbers of $S[1\ldots n]$ are distinct. Denote by $S'[1\ldots n]$ the sorted sequence of $S[1\ldots n]$.

   (a) Let $p_i = P(S'[i] = x)$ be the probability that $S'[i]$ happens to be the pivot $x$. Please give an exact formula for $p_i$ as a bivariate function of $n$ and $i$.

   (b) Ideally, the "best" pivot would be $x = S'\left[\left\lfloor\frac{n+1}{2}\right\rfloor\right]$. Please assess the improvement upon the likelihood of choosing the best pivot, compared to the ordinary version.

   (c) Suppose that an "admissible" pivot $x$ ranges from $S'\left[\left\lfloor\frac{n}{4}\right\rfloor\right]$ to $S'\left[\left\lfloor\frac{3n}{4}\right\rfloor\right]$. Please assess the improvement upon the likelihood of choosing an admissible pivot, compared to the ordinary version.

3. A permutation of sequence [1, 2, … , n] is stack-valid if it can be generated using the following procedure:

   - There is an initially empty stack $S$.
   - $n$ push and $n$ pop operations are intermixed and performed on $S$.
   - The $n$ push operations push the numbers 1 to $n$ sequentially to $S$.
   - A number is printed when popped out of $S$.
   - The stack is empty after all push/pop operations.

   For example, [5, 1, 4, 3, 2] is not stack-valid. On the other hand, [1, 3, 4, 5, 2] is stack-valid, since it can be obtained by performing the following stack operations: PUSH 1, POP, PUSH 2, PUSH 3, POP, PUSH 4, POP, PUSH 5, POP, POP.

   (a) Determine whether [3, 2, 4, 1, 5] is stack-valid. If so, write down the exact order of the stack operations that output this sequence (similar to the example shown above). If not, briefly explain why. (10%)

   (b) Given a permutation of sequence [1, 2, … , $n$], describe an algorithm to determine whether it is stack-valid, and give the time complexity of your algorithm with the Θ notation. (20%)

4. (10%) Construct examples where Dijkstra's algorithm works correctly in the presence of a negative edges and where it works incorrectly in in the presence of a negative edges.

5. (10%) What are the differences and similarities between divide-and-conquer and prune-and search?

6. Suppose the data given to Inorder and Postorder are as follows:
   Inorder：3,4,5,6,7,9,10
   Postorder：3,5,6,4,9,10,7

(a) What's the Preorder? (10%)

(b) If given the pre-order and post-order data, can you find the only middle-order data? Why? (10%)