Algorithms CS4120 Homework

**Analyzing algorithms**

True or false: The following operation is a constant-time op 350

* TRUE

True or false: The following operation is a constant-time op A [ i + 2 ] = A [ i ] + A [ i + 1 ]

* TRUE

Consider the INSERTION-SORT discussed in the class. Run the algorithm on input sequence < − 15 , 30 , 0 , 8 , − 20 >.

Follow the operation of the INSERTION-SORT shown in page 5 of this [filePreview the document](https://bgsu.instructure.com/courses/1319824/files/87986560/download?wrap=1). Show the intermediate sequence of array element in the format of <, , , , >.

**Fill only the necessary blank(s). Fill NA in the unnecessary blank(s).**

**The first sequence is given for (a).**

1. <-15, 30, 0, 8, -20>
2. <-15, 30, 0, 8, -20>
3. <-15, 0, 8, 30, -20>
4. <-20, -15, 0, 8, 30>

Consider the selectionSort function below. (The code is borrowed from [here (Links to an external site.)](https://www.geeksforgeeks.org/selection-sort/).)

**void** selectionSort(int arr[], int n)

1 {

2  **int** i, j, min\_idx;

3    // One by one move boundary of unsorted subarray

4    for (i = 0; i < n-1; i++)

5    {

6        // Find the minimum element in unsorted array

7        min\_idx = i;

8        for (j = i+1; j < n; j++)

9          if (arr[ j ] < arr[ min\_idx ]

10            min\_idx = j;

11        // Swap the found minimum element with the first element

12       swap(&arr[ min\_idx ], &arr[ i ]);

13    }

14 }

Fill in the blanks to complete the corresponding pseudocode to the C++ code above.

* For the sake of time, you may remove the comment in the C++ code.
* **Use \id  (followed by a white space) to indicate one level of indentation**
  + Example: \id \id if x = 1 means the if statement follows two indentations.
* Use **// Swap** as the pseudocode corresponding to line 11 ~ 12 of the C++ code.
* Input A is the name of the array. Input n is the length of the array. You may keep using the names of the local variables.
* **Fill ONLY the necessary blanks. Fill NA in unnecessary blank(s).**

for i = 1 to n - 1

 \id min\_idx = i

 \id for j = i + 1 to n

 \id \id if A[ j ] < A[ min\_idx ]

 \id \id \id min\_idx = j

\id swap (A[min\_idx], A[i])

Consider the SelectionSort algorithm in the previous question.

Define the loop invariant of the outermost loop. **This question will be manually graded.**

For your reference, the original C++ code is pasted below.

**void** selectionSort(int arr[], int n)

1 {

2  **int** i, j, min\_idx;

3    // One by one move boundary of unsorted subarray

4    for (i = 0; i < n-1; i++)

5    {

6        // Find the minimum element in unsorted array

7        min\_idx = i;

8        for (j = i+1; j < n; j++)

9          if (arr[ j ] < arr[ min\_idx ]

10            min\_idx = j;

11        // Swap the found minimum element with the first element

12       swap(&arr[ min\_idx ], &arr[ i ]);

13    }

14 }

Subarray 𝐴[1 . . 𝑖 − 1] contains (𝑖 − 1)th smallest element(s) the of the input sequence and they are sorted.

Consider the running time function T ( n ) of the INSERTION-SORT algorithm.

* tj= 1 in the best-case scenario
* tj= j in the worst-case scenario

Consider the pseudocode shown below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pseudocode | Cost | Time |
| 1 | **for** j = 1 **to** n | c 1 | n |
| 2 | k = j |  |  |
| 3 | **while** k < n |  |  |
| 4 | k+=3 |  |  |

Task 1: In the textbox below, complete the **Cost** and **Time** columns in the format of line number, Cost, Time

* The **Cost**and **Time** column for line 1 should be written as 1, c 1, n
  + Use MATH Editor for mathematical expressions such as c i, n,
    - You may use tj to indicate the number of executions of the **while**-loop for a value of j.

Task 2: After completing the **Cost**and **Time** columns, derive the running time function T ( n ) of the pseudocode in terms of n and tj .

Task 3: Give an expression of Σ tj in terms of n.

Task1:

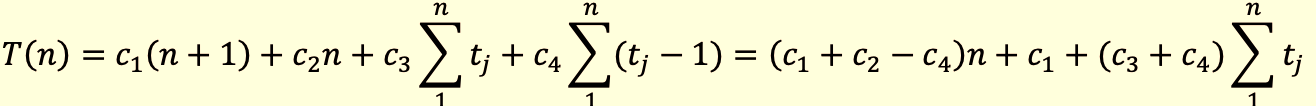
1, 𝑐1, 𝑛 + 1

2, 𝑐2, 𝑛

3, 𝑐3, ∑𝑡j

4, 𝑐4, ∑ (𝑡j −1)

Task2:



Task3:

A close up of a clock

Description automatically generated

# Growth Functions

Is 22n = O ( 2n )? Justify.

No, Assume 22n = 𝑂(2n). The 𝑓(𝑛) = 22n and 𝑔(𝑛) = 2n

Then by the definition of the 𝑂-notation, there exist