Student Name: .... **ID:** ....

Part A [**32 Points**] There are 8 multiple-choice questions in this part.
To choose an answer, simply draw a circle around the bullet (•) of the answer chosen (if any!).
A correct choice for one question will get **4** point. An incorrect answer or marking several answers for a question will get -1. The minimum total mark for this part is 0.

* An Algorithm C1(n) uses $10n \log_2 n$ operations and C2(n) uses $n^2 + 512$ operations. Which of the following values for n_0 is the **smallest** such that C2(n) $\geq$ C1(n) for all $n \geq n_0$?

• 64 • 128 • 16 • 32

* Consider the following pseudo code:

a ← 0
for i ← 0 **to** n∗n **do**
  for j ← 0 **to** i **do** a ← a + i

Which of the following characterization, in terms of n, of the running time of the above code is **not** correct?

• $\Omega(n^2)$ • $O(n^5)$ • $\Theta(n^4)$ • $O(n^3)$

* Assume the Sequence ADT based on a circular array implementation. Which of the following operations requires $O(n)$ steps?

- atRank
- replaceAtRank
- remove
- elemAtRank

* Consider a sequence S implemented with a non-circular array. Suppose further that $S = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$, in that order, so that each element is actually equal to its rank. Which of the following statements about elemAtRank and insertAtRank is **correct**?

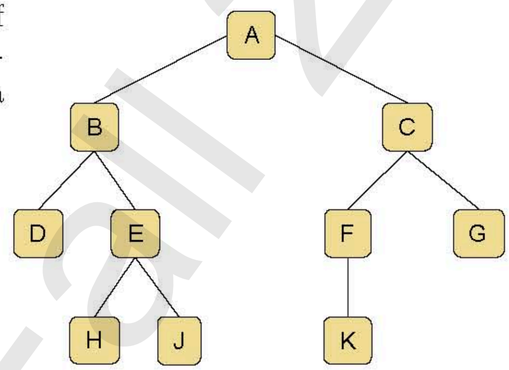
- S.remove(9) has the longest runtime
- S.elemAtRank(9) has the longest runtime
- S.insertAtRank(0,11) has the longest runtime
- S.insertAtRank(0,11) has the shortest runtime

Student Name: .... **ID:** ....

* Consider the following List ADT operations (where the p_i's represent positions): p_1=insertFirst(8), p_2=insertAfter(p_1,5), p_3=insertBefore(p_2,3), p_4=insertFirst(9), remove(p_4), swapElements(p_1, p_2), replaceElement(p_3, 7), p_5=insertAfter(first(),2). Which of the following options describes the final list in a **correct** way?

- (8,5,3,9)
- (5,7,3,2)
- (5,2,7,8)
- (3,7,5,9)

* Which of the following chronological sequences of node names does **not** correctly represents a traversal of the tree to the right that corresponds to a preorder or inorder or postorder traversal?

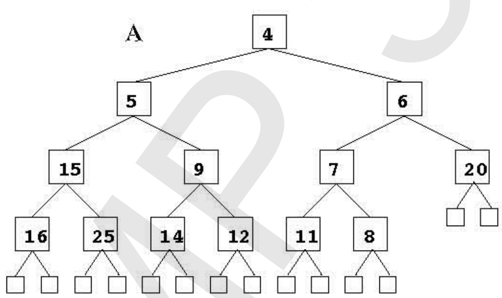


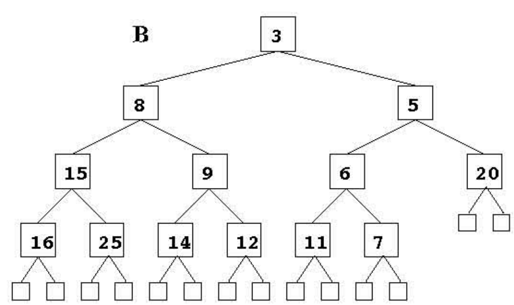
- A,B,D,E.H,J,C,F,K,G
- D,H,J,E,B,K,F,G,C,A
- A,B,E,H,J,K,D,C,F,G
- D,B,H,E,J,A,K,F,C,G

* Which of the following options list the functions in **correct** non-decreasing order if they are compared by asymptotic growth?

- $\log n, n, n\log n, n^2 + \log n, n - n^3 + 7n^5, 2^n, n^n, n!$
- $\log n, n, n\log n, n^2 + \log n, n - n^3 + 7n^5, 2^n, n!, n^n$
- $\log n, n, n\log n, n^2 + \log n, n - n^3 + 7n^5, n!, 2^n, n^n$
- $\log n, n, n\log n, n - n^3 + 7n^5, n^2 + \log n, 2^n, n^n, n!$

* What is the **correct** sequence of `insert` and/or `removeMin` operations on heap A that will transform it into heap B (as shown below)?





- `insert(7),removeMin()`
- `removeMin(), removeMin(), insert(3)`
- `removeMin(), insert(3)`
- `removeMin(), insert(3), insert(5)`

Student Name: .... **ID:** ....

Part B [44 Points]

B.1 [**20 Points**] Write in pseudo code a tail-recursive algorithm "reverse" using the Sequence ADT that creates from a given sequence a new copy in reverse order. For instance, if S1 represents the sequence (1,2,3), then S2 := reverse(S1) returns a sequence that represents (3,2,1) but leaves S1 unchanged.

1. [**8 Points**] Describe in pseudo-code a tail-recursive version of reverse.

2. [**8 Points**] Describe in pseudo-code an iterative version of reverse.

3. [**4 Points**] What are the worst-case runtimes of your above algorithms (use the big-Oh notation). Justify your answers.

Student Name: .... ID: ....

B.2 [**24 Points**] The *Towers of Providence* is a variation of the classical Towers of Hanoi problem. There are four pegs, denoted A, B, C, and D, and n disks of different sizes. Originally, all the disks are on peg A, stacked in decreasing size from bottom to top. Our goal is to transfer all the disks to peg D, and the rules are that we can only move one disk at a time, and no disk can be moved onto a smaller one. Write a recursive algorithm in **pseudo code** that solves the *Towers of Providence* problem.
Hint: use the operator '&' to concatenate elements and lists, the functions first(L) which returns the first element of a list L, rest(L) which return the list L without its first element, and the predicate empty(L) which is true if L does not contain an element.

Student Name: .... **ID:** ....

Part C [**24 Points**] For each of the 4 questions in this part, mark **T** if the given statement is **ALWAYS** true. Otherwise mark **F** and **justify** your answer. If you do not justify the FALSE case you will lose marks. There is **no penalty** for selecting the wrong answer.
Hint: A correct counter example or the correct specification will give you the full mark.

- If $f(n)$ and $g(n)$ are both $O(h(n))$, then $f(n) + g(n)$ is **not** $O(h(n))$. ☐ **T** ☐ **F**

- If x and y are real numbers such that $0 < x < y$ then n^x is $O(n^y)$ and n^y is $O(n^x)$. ☐ **T** ☐ **F**

- A preorder traversal of a heap yields the keys in nondecreasing order. ☐ **T** ☐ **F**

- if $f(n)$ is $\Omega(n)$ and $O(n)$, then $f(n)$ is $\Theta(n)$. ☐ **T** ☐ **F**