York University
EECS 2011Z Winter 2015 – Midterm Tuesday Feb 24
Instructor: James Elder

1. ($4 \times 2 = 8$ marks) What are the possible position indices of all grandchildren of a node with position index n, in an array-based implementation of a binary tree?

   - Answer: $4n, 4n+1, 4n+2, 4n+3$

2. (4 marks) Given an array-based min heap with at least 5 entries, describe in one sentence an $\mathcal{O}(1)$ method for modifying the min heap so that it remains a min heap containing exactly the same entries, but with two of the entries in different positions.

   - Answer: Swap any two sibling leafs. For example, if there are n entries, swap entries $2\lceil n/2 \rceil - 2$ and $2\lceil n/2 \rceil - 1$.

3. ($4 \times 2 = 8$ marks) What are the two main strategies that may be used to implement an iterator? Name one disadvantage of each.

   - Answer:
     Strategy 1: Copy the collection. Disadvantage: wastes time and memory.
     Strategy 2: Use the collection itself. Disadvantage: if the collection is changed results may be unpredictable.

4. ($2 \times 3 = 6$ marks)Algorithm A is $\Omega\left(n^2\right)$ and Algorithm B is $\mathcal{O}(n \log n)$. If (worst-case) asymptotic run time is your only concern, which algorithm should you choose? Is this guaranteed to be the right choice?

   - Answer: One should choose Algorithm B, since its run time is guaranteed to be $n \log n$ at worst, whereas Algorithm A cannot be better than n^2, which is slower. This is guaranteed to be the right choice.

5. ($2 \times 6 = 12$ marks) State the definition of $\mathcal{O}()$ and use it to show that $f(n) = 3n^3 + 5n^2 \log n + 9\cos(2\pi n)$ is $\mathcal{O}\left(n^3\right)$.

   - Answer:

   $$f(n) \in \mathcal{O}\left(g(n)\right) \iff \exists n_0, c : f(n) \leq cg(n) \forall n \geq n_0.$$

   Let $n_0 = 1$. Note that $\forall n \geq n_0$, $\log n \leq n$ and $\cos(2\pi n) \leq n^3$. Thus we have that

   $$f(n) \leq 3n^3 + 5n^3 + 9n^3 = 17n^3, \forall n \geq n_0.$$

   Choosing $c = 17$, we have that $f(n) \leq cn^3 \ \ \forall n \geq n_0$ and therefore that $f(n) \in \mathcal{O}(g(n))$.

6. ($10 \times 1 = 10$ marks) Suppose that class **InPatient** is a subclass of class **Patient**. In the table below, identify the type and outcome of the conversions on lines 5 - 9 of the following code fragment:

```
1        Patient patientA = new Patient();
2        Patient patientB = new InPatient();
3        InPatient patientC = new InPatient();
4
5        patientC = (InPatient) patientA;
6        patientA = patientC;
7        patientC = (InPatient) patientB;
8        patientC = patientB;
9        patientC = patientA;
```

- Answer:

Line	Type of Conversion (Widening or Narrowing)	Outcome (Valid, Compile error, Possible run-time error)
5	Narrowing	Possible run-time error
6	Widening	Valid
7	Narrowing	Valid
8	Narrowing	Compile error
9	Narrowing	Compile error

7. ($8 \times 3 = 24$ marks) What would be your first choice ADT and concrete data structure for the following applications, based on asymptotic run time? In each case, state the asymptotic run time for the indicated add and remove operations, given n entries in the database.

   (a) Fair management of a collection of calls on a call-in radio program. Here"fair" means that if Call A arrives before Call B, Call A is served before Call B.

      i. ADT:

         - Answer: FIFO Queue

      ii. Concrete data structure:

         - Answer: Circular array

      iii. Asymptotic run time for adding a call to the collection:

         - Answer: $\Theta(1)$

      iv. Asymptotic run time for selecting and removing the next call from the collection:

         - Answer: $\Theta(1)$

   (b) Sequencing a collection of customers at airport check-in to ensure as few as possible miss their flight. Assume that customers arrive individually and are travelling on a variety of different flights.

      i. ADT:

         - Answer: Priority Queue

      ii. Concrete data structure:

         - Answer: Heap

      iii. Asymptotic run time for adding a customer to the collection:

         - Answer: $\Theta(\log n)$

      iv. Asymptotic run time for selecting and removing the next customer from the collection:

- Answer: $\Theta(\log n)$

8. (28 marks) You are designing a *recursive* algorithm that will be part of a royal family ancestry database. Specifically, your algorithm **printMatrilinealDescent(person)** will output the maternal descent (ancestry) of the specified royal person in chronological order (i.e., from earliest to latest), starting with the earliest known female ancestor. For example, if **person** represents Queen Elizabeth II of England, **printMatrilinealDescent(person)** will output:

   Mary Garritt →Frances Webb → Anne Salisbury → Caroline Burnaby → Cecilia Cavendish-Bentinck → Elizabeth Bowes-Lyon → Elizabeth II,

   where Elizabeth Bowes-Lyon is Elizabeth II's mother and Cecilia Cavendish-Bentinck is her grandmother, etc.

   Your algorithm can make use of the following pre-existing methods:

   mother(person): returns the mother of the specified person. Returns null if the mother is unknown.
   print(person): prints the name of the person.
   printRightArrow(): prints a right arrow.

   State your algorithm below in concise, clear pseudocode. The algorithm *must* be recursive.

   Algorithm printMatrilinealDescent (person)

   - Answer:

```
if person = null then return
else
    if mother(person) = null then
        print(person)
    else
        printMatrialinealDescent(mother(person))
        printRightArrow()
        print(person)
    end if
end if
```