

## Module 2 Quiz

Quiz, 10 questions

1  
point

1.

Consider the following parallel pseudo-code that use future tasks, as introduced in Lecture 2.1. Select the statement below which must be true for this code snippet.

```
1  finish {  
2    future A = future { S1; return 1;} ;  
3    future B = future { S2; int a = A.get(); S3; return 3;} ;  
4    future C = future { int a = A.get(); S4; int b = B.get();  
5                      S5; return 5;} ;  
6 }
```

- ☐ A. S1 can run in parallel with S3.
  - ☐ B. S2 can run in parallel with S5.
  - ☐ C. S1, S2, and S4 can all run in parallel with each other
  - ☒ D. S2 can run in parallel with S4.
  - ☐ E. S4 can run in parallel with S5.
-

1

point

## Module 2 Quiz

Quiz 210 questions

Which of the following dependencies result from the following pseudo-code using futures (as introduced in Lecture 2.1)?

```
1  finish {  
2      future A = future { int a = 1; S1; S2; return a;} ;  
3      future B = future { int b = 2; S3; int a = A.get(); S4;  
4                          return a+b;} ;  
5  }
```

- ☐ A. S3 depends on S2 having executed
  - ☒ B. S4 depends on S2 having executed
  - ☐ C. S2 depends on S4 having executed
  - ☐ D. S1 depends on S3 having executed
- 

1

point

3.

You can use futures in the Java Fork-Join framework (as discussed in Lecture 2.2) by implementing which class?

- ☐ A. RecursiveAction
- ☒ B. RecursiveTask



## Module 2 Quiz

Quiz, 10 questions

---

1

point

4.

How do you retrieve the value from a future in the Java Fork-Join framework (as discussed in Lecture 2.2) ?



A. Through the return value provided by explicitly calling a task's compute() method.



B. By the return value of the fork() method of ForkJoinTask.



C. By the return value of the join() method of ForkJoinTask.



D. By the return value of the quietlyJoin() method of ForkJoinTask.

---

1

point

5.

Consider the Pascal's triangle implementation below, with a triangle containing R rows, with row  $i$  containing  $i$  entries. A triangle with R rows would therefore have  $\frac{R(R+1)}{2}$  total entries. Assuming a memoized parallelization strategy like the one below (and discussed in Lecture 2.3), how many futures would be created when computing Pascal's triangle with R rows ?

```
1 final Map<Pair<Int, Int>, future<Integer>> cache = new ...;
2 future<Integer> choose(final int N, final int K) {
3     final Pair<Int, Int> key = Pair.factory(N, K);
4     if (!cache.contains(key)) {
5         future<Integer> newFuture = null;
6         if (N == 0) {
7             ...
8         }
9     }
10 }
```

## Module 2 Quiz

Quiz, 10 questions

```
7         // Root element case, zero gets
8         newFuture = future {
9             return 1;
10        }
11    } else if (K == 0) {
12        // Left edge element case, one get
13        newFuture = future { return choose(N - 1, K).get(); }
14    } else if (N == K) {
15        // Right edge element case, one get
16        newFuture = future { return choose(N - 1, K - 1).get(); }
17    } else {
18        // Inner element case, two gets
19        newFuture = future {
20            return choose(N - 1, K - 1).get() +
21                choose(N - 1, K).get();
22        }
23    }
24    cache.put(key, newFuture);
25 }
26 return cache.get(key);
27 }
28 // Scan across target row of Pascal's triangle
29 final int targetRowLength = targetRowIndex + 1;
30 for (int i = 0; i < targetRowLength; i++) {
31     result[i] = choose(targetRowIndex, i).get();
32 }
33
```

- ☐ A.  $R$
- ☐ B.  $2 \times R$
- ☐ C.  $R(R + 1)$
- ☒ D.  $\frac{R(R+1)}{2}$

---

## Module 2 Quiz

Quiz, 10 questions

6.

Based on the same Pascal's triangle implementation above, if memoization and futures are used to compute a Pascal's triangle of  $R$  rows, how many future `get()` calls must be made to compute the values for the  $R$  rows? Keep in mind the special cases of elements on the left and right edges of the triangle. You should ignore the `get()` calls on the second to last line (line 31) in the code example above; only consider the `get()`s necessary to compute the actual contents of the triangle.

- ☒ A.  $R^2 - R$
- ☐ B.  $\frac{R(R+1)}{2}$
- ☐ C.  $R(R + 1)$
- ☐ D.  $2 \times R$
- 

1  
point

7.

Supposed you had a List of Integers in Java: [3, 6, 8, 2, 1, 0]. Which of the stream-based programs below would be equivalent to the provided loop-based Java code? (Recall that Java streams were introduced in Lecture 2.4.)

## Module 2 Quiz

Quiz, 10 questions

```
1 List<Integer> output = new List<Integer>();
2 for (Integer i : input) {
3     if (i >= 3) {
4         output.add(i);
5     }
6 }
```

- ☐ A. input.filter(v -> v >= 3);
- ☒ B. input.stream().filter(v -> v >= 3);
- ☐ C. input.stream().filter(v -> v < 3);
- ☐ D. input.stream().average();

1  
point

8.

Consider a search algorithm that uses many tasks to examine the search space in parallel. Every task that discovers a target value in the search space will set a global boolean flag to true (it is initialized to false). However, no task will ever read this global flag during execution, hence there will be no early exit of tasks. The flag will only be read after all tasks have terminated.

Such a program will exhibit which of the following? (Recall that data races, functional determinism, and structural determinism were introduced in Lecture 2.5.)

- ☐ A. Functional and structural determinism, and no data race
- ☒ B. Functional and structural determinism, with a data race



☐ C. Functional but not structural determinism, with a data race

## Module 2 Quiz

☐ D. Structural but not functional determinism, with a data race

Quiz, 10 questions

---

1  
point

9.

Now consider another search algorithm that uses many tasks to examine the search space in parallel. Each task increments a shared integer counter (e.g., `count = count + 1`) when the search is successful.

Such a program will exhibit which of the following?

- ☐ A. Functional and structural determinism, and no data race
  - ☐ B. Functional and structural determinism, with a data race
  - ☐ C. Functional but not structural determinism, with a data race
  - ☒ D. Structural but not functional determinism, with a data race
- 

1  
point

10.

Finally, consider a variation of Question 8 in which every task that discovers a target value in the search space will set a shared global int variable to the location of the target value that was found (the variable is initialized to -1). However, no task will ever read the int variable during execution, hence there will be no early exit of tasks. The variable will only be read after all tasks have terminated. Note that, in general, there can be multiple possible locations for the target value, and we assume that any target location is acceptable in the final value of the int variable.

## Module 2 Quiz

Quiz, 10 questions

Such a program will exhibit which of the following?

- ☐ A. Functional and structural determinism, with a data race
- ☐ B. Functional and structural determinism, and no data race
- ☒ C. Benign non-determinism, with a data race.



I, **Fan Yang**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

[Learn more about Coursera's Honor Code](#)

Submit Quiz

