HackerRank For Work

Fall_2017_INFO6205: Final ... 90 minutes

This test has been locked by r.hillyard@northeastern.edu.

Question - 1

SCORE: **5 points**

Bags

Which of the following is an appropriate application of the *Bag* data structure?

- All elements in a bag must have different keys.
- A bag is a good choice to store the adjacent vertices of a vertex in a graph.
- The order in which elements are added or removed to/from a bag is significant.
- A bag cannot be implemented as a linked list.

Question - 2

SCORE: **5 points**

Red-black trees

Which of the following statements about a red-black tree are true?

- A red-black tree is a type of balanced search tree.
- A red-black tree of order 3 is logically equivalent to a 2-3 binary tree.
- A red-black tree of order 4 is logically equivalent to a B-tree of order 4
- All of the above

Question - 3

SCORE: **10 points**

Solve a problem for a colleague

Sometimes you will ask a colleague to help you with a problem and, as you are explaining the problem to him/her, the light bulb in your brain switches on and you no longer need their help.

Remember how I emphasized that algorithms have nothing to do with any specific language? There are certain principles that transcend the choice of language. A junior colleague, who writes all his code in Scala has come to show you some code that is performing poorly. It is part of a Genetic Algorithm framework and this particular code has to do with females choosing the most attractive male from a "lek" (https://en.wikipedia.org/wiki/Lek_mating).

He thinks the problem is in the following method:

```
def bestMate(female: Y): Y = (males map { m =>
(m, mateable.attraction(female, m))
}).sortBy(_._2).head._1
```

You know enough Scala to know that the *_1* method yields the first element of a tuple, *_2* the second, etc. And of course *head* yields the first element in a list.

The code works perfectly, but is slow and your colleague comes to you as the local "algorithms" expert.

(A) Explain to your colleague what he's doing wrong.

(B) Assuming that, on average, there are 1000 males in the "lek", by what factor do you expect to speed up this bit of the code?

Question - 4

SCORE: **5 points**

Hash Table

Implementing a symbol table by hashing is one of the most prevalent methods. Which of the following statements about a hash table are true?

- [x] Separate chaining is a method which can be reasonably efficient whether a table is fairly empty or almost full, but it tends to be somewhat wasteful of space.
- [x] Linear probing is a method which can be very efficient for low density tables but which gets very inefficient when the table gets full.
- [ ] The hash function must be chosen such that it is impossible to have collisions.
- [x] It is desirable that the hash function distributes the keys uniformly.
- [x] A hash function is, ideally, suited to its key as a hash function which performs well on, say, Double values might not perform so well as a hah function on dates.
- [ ] Like a balanced search tree, it is important that the key to be hashed should have an order defined.
- [x] The standard Java hashing function (hashCode) must be consistent with, i.e. based on the same "primary key" as, the equals method.
- [ ] One advantage of a Hash Table (over other symbol tables) is that it never has to be resized.

Question - 5

SCORE: **6 points**

Quicksort vs. Mergesort

It's tempting to think that the only difference between merge sort and quick sort is that in one case we process the whole first, then partition and recursively work on the partitions; while in the other case, we do it the other way around.

But there's a deeper difference than that, at least in my opinion. And it is essentially the reason why merge sort is well-behaved and predictable in

its performance, while quick sort is not.
Imagine you are at an interview and briefly describe this significant difference. Remember: deep breath... think... then speak/write.

[Just like at an interview, I'm not looking for an essay. Express the difference as succinctly as possible.]

Question - 6

SCORE: **5 points**

Graph data structure (part one)

There are three obvious ways to store graph information:

1. a list of edges
2. a vertex-by-vertex matrix
3. a list of vertices, each with its adjacency list

Assuming a typical graph where $E = cV^2/2$ [E is the number of edges, V is the number of vertices, and c is some number ($c < 1$) that represents the average density of the graph and is typically much smaller than 1], order these three data structures according to the efficiency of space utilization (starting with the most efficient):

- ○ 1, 2, 3
- ◉ 1,3,2
- ○ 2,1,3
- ○ 3,1,2
- ○ 2,3,1
- ○ 3,2,1

Question - 7

SCORE: **5 points**

Graph data structure (part two)

Given the same conditions as in the previous question, order the methods according to the number of operations required to determine if two vertices are connected by an edge, starting with the fewest.

- ○ 1,2,3
- ○ 1,3,2
- ○ 2,1,3
- ○ 3,1,2
- ◉ 2,3,1
- ○ 3,2,1

Question - 8

SCORE: **20 points**

Implement Hashing with Linear Probing

Write the code to implement a linear-probing hash table.

```
The first line of the input is the "k factor"
according to which you will have linearly probe.
The second line represents the number of
operations
The subsequent lines represent the put operation
for the key, value pair.


Sample Input:
3// k factor
3 // number of operations
1,3 //(key,value)
2,3
4,5


Sample Output:
null 3 3 null 5 null null null null null
```

Question - 9

SCORE: **10 points**

Hash function

The Commonwealth of Massachusetts is creating a software system to maintain a web site for each of the 351 cities and towns in the state, keyed by 5-digit zip code (don't worry about the fact that many of them have multiple zip codes--just assume that there's only one zip per town/city). The idea is to use separate chaining with 128 buckets. There are 100,000 possible 5-digit zip codes which will require something like 17 bits. So, the programmer decides to simply strip off the least-significant 10 bits in order to get the hash.
Explain why this is a bad idea, giving a short analysis of the problem/solution (no more than a few lines of text).
To help you with this question (since you are obviously not as familiar with MA zip codes as I am) here is a representative sample of MA zip codes:

01001	Agawam
01007	Belchertown
01013	Chicopee
01033	Granby
01060	Northampton
01085	Westfield
01103	Springfield
01230	Great Barrington
01453	Leominster
01602	Worcester
01741	Carlisle
01854	Lowell
02108	Boston
02138	Cambridge
02360	Plymouth
02493	Weston

02780 Taunton