EC 504 - Advanced Data Structures

Spring 2016

Homework 2, Due Feb 28 (11pm)

Nuts and bolts, 20 pts

You are given n nuts and n bolts such that each nut fits exactly one bolt. Your only means of comparing these nuts and bolts is with a function TEST(x,y), where x is a nut and y is a bolt. The function returns either “nut is too big”, “nut is too small”, or “nut fits perfectly”.

Devise and analyze an O(n^2) worst-case algorithm for sorting the nuts and bolts in order from smallest to largest. Provide your algorithm as Python code.

Finding the median, 30 pts

You are given two sorted arrays A, and B of integer values, in increasing order, of sizes n and m respectively, and m+n is an odd value (so we can define the median value uniquely). Develop an algorithm with worst case complexity of O(log(m + n)) for finding the median of the combined sorted arrays. Partial credit will be given for algorithms which do not meet the required complexity.

You must implement the function in C++.

Asymptotics, 25 pts

Place the following functions from asymptotically smallest to largest. When two functions have the same asymptotic order, put an equal sign between them. Provide an explanation for your ordering.

$$1, n^3, n^{n^n}, n^{\frac{1}{n}}, 0, \frac{n}{4}, n^9+n+2, \sqrt[3]{n}, (n+1)^n, \sum_{k=1}^{\log n}\frac{n}{3^k}, (1+\frac{1}{n})^n, \prod_{k=1}^{n}(1-\frac{1}{k^3}), \log n$$

Recurrences, 25 pts

For each of the following functions, provide:

1. A recurrence $T(n)$ that describes the worst-case runtime of the function in terms of n as provided (i.e. without any optimizations)
2. The tightest asymptotic upper and lower bounds you can find for $T(n)$

```
def A(n):
   if (n == 0):
    return 1
   else:
    return A(n-1) * A(n-1) * A(n-1)
```

```
def B(n):
   if (n == 0):
     return 1
   if (B(n//2) >= 5):
      return B(n//2) + 5
   else:
      return 5
```

```
def C(n):
   if (n <= 1):
    return 1
   sum=0
   for ii in range(int(math.sqrt(n))):
      sum += C(int(math.sqrt(n)))
   return sum
```

```
def D(n):
   if (n <= 1):
    return 1
   count = 1
   tmp = D(n//2)
   for jj in range(n):
      ii=1;
      while (ii<n):
         if (tmp < math.exp(ii+jj)):
            count += 1
         ii*=2
   return D(n//2) * (count % 2)
```

```
def E(n):
    if (n == 0): return 1
    if (n == 1): return 3
    return E(n-1) + E(n-2)*E(n-2)
```

Submission Guidelines

Submit the following files:

- `nuts_and_bolts.py`
- `finding_median.cpp`
- `hw2_analysis.md`

using this link

Each file must start with the following lines:

```
# AUTHOR1: name1 email1@bu.edu
# AUTHOR2: name2 email2@bu.edu
# AUTHOR3: name3 email3@bu.edu
```

Note:

- use # for python, // for C++, and % for Markdown (so that these are comments)
- you must use your bu email as that is the unique key I will use to figure out who is who.
- use as many lines as there are collaborators.