HW 1

Monday, January 17, 2022 5:14 PM

Part (a)

```
void f1(int n)
{
  int t = sqrt(n);
  for(int i = 0; i < n; i++){
    for(int j = 0; j < n; j++){
      // do something O(1)

    }
    n -= t;
  }
}
```

$\sqrt{n}$ $\sqrt{n}$

$$\sum_{i=0}^{\sqrt{n}}\left(\sum_{i=0}^{\sqrt{n}}\theta(1)+\theta(1)\right)=\sum_{i=0}^{\sqrt{n}}\theta(1)+\sum_{i=0}^{\sqrt{n}}\sum_{i=0}^{\sqrt{n}}\theta(1)$$

$$=\sqrt{n}+\sqrt{n}\cdot\sqrt{n}=\theta(n)$$

$\theta(\sqrt{n}\cdot\sqrt{n})$

$\theta(n)$

Part (b)

```
Assume A is an array of size n+1.

void f2(int* A, int n)
{
  for(int i=1; i <= n; i++){
    for(int k=1; k <= n; k++){
      if( A[k] == i){
        for(int m=1; m <= n; m=m+m){
          // do something that takes O(1) time
          // Assume the contents of the A[] array are not changed
        }
      }
    }
  }
}
```

n n

$$\sum_{i=1}^{n}\left(\sum_{i=1}^{n}\left(\sum_{i=1}^{\log n}\theta(1)\right)\right)=\sum_{i=1}^{n}\sum_{i=1}^{n}\theta(\log n)=\sum_{i=1}^{n}\theta(n\log n)=\theta(n^2\log n)$$

assume if statement is always ran for worst case

so the most nested for loop runs everytime

$\log(n)$ because m goes $m=1$ then $m=2$; $m=4$; $m=8$; $m=16$ ...

so $\theta(n^2\log(n))$

Part (c)

```
void f3(int* A, int n)
{
  if(n <= 1) return;
  else {
    f3(A, n-2);
    // do something that takes O(1) time
    f3(A, n-2);
  }
}
```

if $n=10$ then it runs 5 times $\frac{n}{2}$

$n=50$ it runs 25 times

$\frac{n}{2}+\frac{n}{2}=n$ $\theta(n)$

Part (d)

Notice that this code is very similar to what will happen if you keep inserting into an ArrayList (e.g. `vector`). Notice that this is NOT an example of amortized analysis because you are only analyzing *1* call to the function `f()`. If you have discussed amortized analysis, realize that does NOT apply here since amortized analysis applies to *multiple* calls to a function. But you may use similar ideas/approaches as amortized analysis to analyze this runtime. If you have NOT discussed amortized analysis, simply ignore it's mention.

```
int f (int n)
{
  int *a = new int [10];
  int size = 10;
  for (int i = 0; i < n; i ++)
    {
      if (i == size)
        {
          int newsize = 4*size;
          int *b = new int [newsize];
          for (int j = 0; j < size; j ++) b[j] = a[j];
          delete [] a;
          a = b;
          size = newsize;
        }
      a[i] = i*i;
    }
}
```

n

$$\sum_{i=0}^{n}\left(\theta(1)+\sum_{j=0}^{size}\theta(1)\right)=\sum_{i=0}^{n}\theta(1)+\sum_{i=0}^{n}\log_4(n)=n+n\log(n)$$

$\theta(n\log n)$

for $n=1,2,\ldots,10$

$\theta(n)$

for $n=11$ inner loop runs

sets size $=4\cdot$size

so runs at 10, 40, 160, 640