Homework #1 for Algorithms

Due: Friday, Sept. 9 by 3 PM

Chapter 1 problems: 4, 11, 13, 22, 25, and the problems below:

Appendix B problems: 15a, b, c and 19a, b

A) Analyze the below algorithm to determine the obvious big-oh (upper-bound), then trace the algorithm to determine its theta notation, \( \theta() \).

```plaintext
i = n
while ( i > 0 ) {
    for ( j = 1; j < i; j++ ) {
        < something of \( \theta(1) \)>
    } // end for
    i =\lfloor i / 2 \rfloor
} // end while
```

B) Analyze the below algorithm to determine the obvious big-oh (upper-bound), then trace the algorithm to determine its theta notation, \( \theta() \).

```plaintext
for (i = 1;  i <= n;  i++) {
    for (j = 1;  j <= i;  j++) {
        for (k = j;  j <= (i + j);  k++)  {
            <some code that takes \( \theta(1) \) time>
        } // end for k
    } // end for j
} // end for i
```

C) Analyze the below algorithm to determine the obvious big-oh (upper-bound), then trace the algorithm to determine its theta notation, \( \theta() \).

```plaintext
result = 0
for (r=1; r <= n; r++) {
    for (c=1; c <= n*n*n; c++) {
        for (d=1; d<=n; d++) {
            <some code that takes \( \theta(1) \) time>
        } // end for d
    } // end for c
} // end for r
```