Step 1. Find $T(N)$

```
int sum=0;
for (int a=N;a>0;a--) {
    for (int b=a;b>0;b--) {
        sum++;
    }
}
```

Line	Code	Cost
1	**int** *sum*=0;	1
2	**int** *a*=*N*;	1
3	**while** (*a*>0) {	$\sum_{a=1}^{N} 1$
4	**int** *b*=*a*;	$\sum_{a=1}^{N} 1$
5	**while** (*b*>0) {	$\sum_{a=1}^{N}\sum_{b=1}^{a} 1$
6	*sum*++;	$\sum_{a=1}^{N}\sum_{b=1}^{a} 1$
7	*b*--;	$\sum_{a=1}^{N}\sum_{b=1}^{a} 1$
8	}	$\sum_{a=1}^{N} 1$
9	*a*--;	$\sum_{a=1}^{N} 1$
10	}	1
Total		$T(N) = 3\sum_{a=1}^{N}\sum_{b=1}^{a} 1 + 4\sum_{a=1}^{N} 1 + 3$
		$\sum_{a=1}^{N} 1 = N$
		$\sum_{a=1}^{N}\sum_{b=1}^{a} 1 = \sum_{a=1}^{N} a = \frac{N(N+1)}{2} = \frac{1}{2}N^2 + \frac{1}{2}N$
Total		$T(N) = \frac{3}{2}N^2 + \frac{11}{2}N + 3$

Step 2. Find $f(N) = N^2$

Step 3. Find c_{min} and c

$$c_{min} = \lim_{n\to\infty}\frac{T(n)}{f(N)} = \frac{\frac{3}{2}N^2 + \frac{11}{2}N + 3}{N^2} = \frac{3}{2} + \frac{11}{2N} + \frac{3}{N^2} = \frac{3}{2}$$

$$c = c_{min} + \Delta = \frac{3}{2} + 1 = \frac{5}{2}$$

Step 4. Find n_0

$$T(n_0) \le cf(n_0)$$
$$\frac{3}{2}n_0^2 + \frac{11}{2}n_0 + 3 \le \frac{5}{2}n_0^2$$
$$n_0^2 - \frac{11}{2}n_0 - 3 \ge 0$$

And the class and the instructor after *some lengthy* debate finally arrive at $n_0 = 6$

Line	Code	Cost
1	**while (*N*>0) {**	k
2	** *N*=*N*/2;**	$2k$
3	**}**	1

$$T(n) = 3k + 1$$

$$n = n_0 = N$$

$$n_{i+1} = \frac{n_i}{2}$$

$$n_k = \frac{n}{2^k} = 1$$

$$2^k = n \Rightarrow k = \log_2 n$$

k is the number of times that $n > 0$ and the number of times the statement ***n*=*n*/2;** is executed.

n	***k***	$\log_2 n$	$\lfloor \log_2 n \rfloor$	**Comments**	$\lfloor \log_2 n \rfloor + 1$
0	0	∞	0		0+1=1 (error)
1	1	0	0	1/2=0	0+1=1
2	2	1	1	2/2=1 1/2=0	1+1=2
3	2	1.58	1	3/2=1 1/2=0	1+1=2
4	3	2	2	4/2=2 2/2=1 1/2=0	2+1=3
5	3	2.32	2	5/2=2 2/2=1 1/2=0	2+1=3
6	3	2.58	2	6/2=3 3/2=1 1/2=0	2+1=3
7	3	2.81	2	7/2=3 3/2=1 1/2=0	2+1=3
8	4	3	3	8/2=4 4/2=2 2/2=1 1/2=0	3+1=4
9	4	3.17	3	9/2=4 4/2=2 2/2=1 1/2=0	3+1=4

…		3.x	3		3+1=4

n	***k***	$\log_2 n$	$\lfloor \log_2 n \rfloor$	**Comments**	$\lfloor \log_2 n \rfloor + 1$
16	5			16/2=8 8/2=4 4/2=2 2/2=1 1/2=0	4+1=5

$n > 0 : T(n) = 3\lfloor \log_2 n \rfloor + 4$

$n = 0 : T(n) = 1$

```
int Empirical(int n)
{    int c=0;
     while (n>0) {     c++;
          n=n/2;       c++; c++;
     }                 c++
     return c;
}

double log2(double x){return log(x)/log(2.0);}
int Analytical(int n){      if (n>0) return 3*(int)floor(log2((double)n))+4; else return 1;}
```