Problems on Complexity Analysis

Find the Time Complexities of the following snippets of code:

(A)

```
/* Assume that rand() takes constant amount of time */
int a = 0, b = 0;
for (int i = 0; i < N; i++) {
    a = a + rand();
}
for (int j = 0; j < M; ++j) {
    b = b + rand();
}
```

O(N+M)

(B)

```
int a = 0, b = 0;
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        a = a + j;
    }
}
for (int k = 0; k < N; k++) {
    b = b + k;
}
```

O(N^2)

(C)

```
int a = 0;
for (int i = 0; i < N; i++) {
    for (int j = N; j > i; --j) {
        a = a + i + j;
    }
}
```

O(N^2)

(D)

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

O(logN)

(E) O(N^K)

```
void fun(int N, int K) {
    for(int i = 1; i <= N; i++) {
        /* Assume that pow() takes constant amount of time */
        int P = pow(i, K);
        for (int j = 1; j <= P; ++j) {
            /* Some constant amount of computation */
        }
    }
}
```

(F) O(NlogN)

```
int count = 0;
for(int i = N; i > 0; i /= 2) {
    for(int j = 0; j < i; j++) {
        count += 1;
    }
}
```

(G) O(NlogN)

```
int k = 0;
for(int i = N/2; i <= N; ++i) {
    for (int j = 2; j <= N; j = j * 2) {
        k = k + N/2;
   }
}
```

(H) O(N)

```
int j = 0;
for(int i = 0; i < N; ++i) {
    while(j < N && arr[i] <= arr[j]) {
        j++;
    }
}
```

Match the following Time Complexities:

(1) Linear	**(A)** N^{K+G}	**(D)**
(2) Logarithmic	**(B)** $5^{N \times 2}$	**(F)**
(3) Exponential	**(C)** $\frac{N}{4}\log_2(\frac{N}{4000})$	**(B)**
(4) Polynomial	**(D)** $3^{20}N + 10^5$	**(A)**
(5) Log Linear	**(E)** $10N + 9\frac{N}{100} + 340N^2$	**(C)**
(6) Quadratic	**(F)** $10^3\log_2(N+3N)$	**(E)**