

Corso di Ingegneria degli Algoritmi
Facoltà di Ingegneria, Sapienza Università di Roma
Appello luglio 2010 (5/6 cfu) - Sessione A (8/7/2010)

Durata: 2 ore

Domanda 1. Si considerino i seguenti due programmi C:

Programma A	Programma B
<pre>#define N ... #include <stdio.h> typedef struct { char a; int b; short c[26]; } T; void foo(T x) { if (x.b < 1) return; x.b--; foo(x); } int main() { T x = { 0, N, 0 }; foo(x); printf("%d\n", x.b); return 0; }</pre>	<pre>#define N ... #include <stdio.h> typedef struct { char a; int b; short c[26]; } T; void foo(T* x) { if (x->b < 1) return; x->b--; foo(x); } int main() { T x = { 0, N, 0 }; foo(&x); printf("%d\n", x.b); return 0; }</pre>

Si assuma che $\text{sizeof}(\text{char})=1$, $\text{sizeof}(\text{short})=2$, $\text{sizeof}(\text{int})=4$ e $\text{sizeof}(\text{void}^*)=4$. Assumendo inoltre che la dimensione massima della pila di esecuzione sia di 64 KB e che ogni record di attivazione di funzione richieda 4 byte oltre allo spazio per variabili locali/parametri formali e per il valore di ritorno della funzione, dire per quali delle seguenti definizioni della macro N i programmi terminano correttamente e cosa stampano:

1. #define N 250
2. #define N 500
3. #define N 2000
4. #define N 4000
5. #define N 12000

Domanda 2.

Si dimostri che il numero minimo di accessi a disco richiesti nel caso peggiore da qualsiasi algoritmo di ordinamento in memoria esterna è $\Omega((N/B) \log_{(M/B)} (N/B))$, dove N è il numero di item da ordinare, B è il numero di item trasferiti fra disco e memoria

interna ad ogni accesso a disco, e M è il numero di item memorizzabili in memoria interna.

Domanda 3.

Si discutano i seguenti meccanismi di base degli allocatori di memoria, spiegando cosa sono, a cosa servono e perché vengono impiegati:

1. header e footer di blocco
2. bounday tags
3. puntatori a blocchi nei blocchi stessi
4. lookup table