

Corso di Ingegneria degli Algoritmi
Facoltà di Ingegneria, Sapienza Università di Roma

Appello del 17 giugno 2010 (5/6 cfu) – Sessione A (21/6/2010)

Durata: 2 ore

Domanda 1.

a) Si scrivano in C le seguenti dichiarazioni:

1. `v` è un puntatore a puntatore a funzione che prende come primo parametro un puntatore a funzione senza argomenti che restituisce `int`, come secondo parametro un `double`, e come terzo parametro un array di puntatori a carattere, e restituisce un puntatore a una funzione che prende come parametro un `int` e restituisce un puntatore a puntatore a `void`;
2. `q` è un array di puntatori a struttura avente come campi un puntatore a funzione senza argomenti che restituisce `void` e un `int`;

b) Si traduca in linguaggio naturale la seguente dichiarazione C:

```
void* (**f[10])(void *(*g)(int (*h)()))
```

Domanda 2.

Si consideri il tipo di dato *array dinamico* con le seguenti operazioni:

1. `expand()`: incrementa di 1 il massimo indice valido dell'array;
2. `shrink()`: decrementa di 1 il massimo indice valido dell'array;
3. `set(i, v)`: assegna il valore `v` alla cella di indice `i` nell'array;
4. `get(i)`: restituisce il valore della cella di indice `i` dell'array.

Si assuma che `n` denoti il numero corrente di elementi dell'array dinamico e che gli indici validi siano compresi tra 0 e `n-1` (inclusi).

Si illustri una realizzazione dell'array in modo che ogni operazione abbia costo costante ammortizzato e lo spazio richiesto sia in ogni momento $O(n)$. Si analizzi il costo delle operazioni utilizzando il metodo dei crediti.

Domanda 3. Si descriva l'algoritmo di fusione a `k` vie per l'ordinamento in memoria secondaria. Usando la notazione asintotica, si analizzi il numero di accessi a disco effettuati dall'algoritmo nel caso peggiore in funzione del numero `N` di elementi da ordinare, il numero di elementi `M` memorizzabili in memoria centrale, e il numero di elementi `B` trasferiti tra memoria secondaria e memoria centrale ad ogni accesso a disco.