Algoritmica 2
Compitino: 17/12/2010
Soluzione

Esercizio 1
Basandosi sulla nozione nota che il problema del Knapsack con profitti interi è NP-hard, dimostrare che il problema del Knapsack non è assolutamente approssimabile se P ≠ NP.

Vedi dispensa su Problemi Difficili, pagine 560-61 e 565.

Esercizio 2

Sia dato l'insieme di chiavi S= {5, 10, 3, 28, 29, 25, 17, 2}. Progettare una soluzione per il problema del Dizionario Statico formulato su S basata su Hash Perfetto, indicando le funzioni hash (universali) adottate con i loro parametri, motivando le scelte fatte, disegnando l'allocazione delle chiavi di S nelle tabelle risultati.

Abbiamo n=8 chiavi in S, per cui definiamo la dimensione della tabella hash di primo livello come m=11, e scegliamo come funzione hash universale <a_1,a_2> = <2,1>, entrambi moltiplicatori delle varie cifre della chiave (come visto in classe). La tabella di primo livello risulta T=[-,-,{2,10,29},3,-,5,-,-,-,{17,25},-,-,-] dove tra parentesi graffe abbiamo indicato i gruppi di chiavi che collidono su una certa cella. Nel caso di assenza di collisioni, costruiamo tabelle di secondo livello costituite da una sola cella, e quindi senza necessità di definire per esse funzioni hash universali. Per quanto riguarda invece le celle 2 e 9, siccome siamo in presenza di collisioni, definiamo due tabelle di secondo livello T_2 e T_9 aventi dimensione m_2 = 11 > (n_2)^2 = 9 e m_9 = 5 > (n_9)^2 = 4. Come funzione hash prendiamo la stessa, ossia h_2 = h_9 = <1,0>, e così otteniamo una distribuzione perfetta pari a T_2 = [2,10,29] e T_9 = [25,17].

Esercizio 3

Si consideri il testo T=T[1,12]=**ABCXABCYABCD** e il pattern P=P[1,4]=**ABCD**. Si indichi:

1) La lista degli shift operati dall'algoritmo di Knuth Morris e Pratt per la ricerca di P in T (ovvero la lista delle posizioni del testo cui si allinea la prima posizione del pattern).
2) La lista completa di coppie (i,j) dei confronti T[i]<->P[j] che l'algoritmo suddetto effettua.

```
1) 1,4,5,8,9
2)(1,1),(2,2),(3,3),(4,4),(4,1),(5,1),(6,2),(7,3),(8,4),(8,1)
(9,1),(10,2),(11,3),(12,4)
```

Esercizio 4
Si consideri una memoria di dimensione k=4 che contiene le pagine **A, B, C, D** e la sequenza di richieste σ = **ABCXABCYABCD,** mostrare la sequenza di pagine eliminate dalla memoria applicando le tecniche:

1) LRU
2) OPT
3) MARKING
4) DG (Dynamic Access Graph) con γ= 2, β= 1.5, α=0.8

1) LRU : D, X, Y
2) OPT: D, X, Y
3) MARKING: D, X, Y
4) DG: All'inizio la memoria contiene: A, B, C, D; dopo le richieste A, B, C, X il grafo contiene gli archi (A, B) , (B,C) e (C,X) tutti a peso 1 e D non è connesso e quindi è a distanza $\propto$ da X. La pagina D viene eliminata. Dopo le richieste A, B, C, Y, abbiamo che la distanza A, Y =2.6, B,Y =1.8 C,Y =1 e XY = 3.6, la pagina X viene eliminata. Poiché γk= 8, tutti gli archi del grafo vanno aggiornati moltiplicandoli per 1.5; abbiamo (A,B)=1.2, (B,C) = 1.2, (C,X) = (C,Y) = (X,A) = 1.5. Dopo le ultime richieste A, B, C, D, la pagina a distanza massima da D è Y che dista 3.92, che viene eliminata.

Esercizio 5
È dato un albero binario di ricerca completo, di altezza pari a 3, che memorizza le seguenti n=15 chiavi (in ordine simmetrico): A, B, C, D, E, F,G, H, I, L, M, N, O, P Q:

1) Mostrare l'ordine con cui vengono allocate tali chiavi in un array T secondo lo schema cache-oblivious di Van Emde Boas (VEB) tree. Motivare la risposta.
2) Supporre che la dimensione del blocco di trasferimento sia b = 2 e che quindi l'array T sia memorizzato in n/b = 8 blocchi consecutivi, in cui tutti i blocchi contengono due chiavi tranne l'ultimo blocco (che ne contiene una sola). Indicare quanti e quali blocchi di T sono acceduti per la ricerca della chiave G nell'albero allocato in T secondo lo schema del punto 1).

L'albero viene diviso in due metà e le chiavi vengono ricorsivamente allocate: prima quelle dell'alberino superiore e poi quelle degli alberini inferiori (nell'ordine da sinistra a destra). Il caso base di una sola chiave è immediato.

1) Nell'esempio, l'alberino superiore contiene le chiavi {D, H, N} e i quattro alberini inferiori contengono rispettivamente {A,B,C}, {E,F,G},{I,L,M} e {O,P,Q}. Poichè ciascuno di questi alberini contiene tre chiavi{x,y,z}, l'ordine imposto dal VEB tree è y x z. In conclusione, la risposta è
T = H D N B A C F E G L I M P O Q

2) La memorizzazione di T a blocchi induce la seguente partizione:
T = [H D] [N B] [A C] [F E] [G L] [I M] [P O] [Q]

La ricerca della chiave G confronta H, D, F, G con tale chiave: richiede quindi di accedere ai tre blocchi [*H D*], [*F* E] e [*G* L].