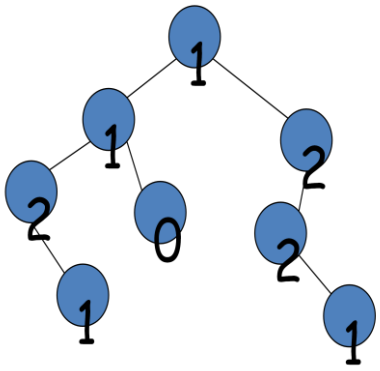
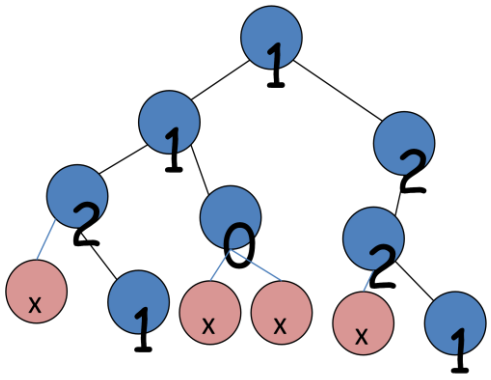
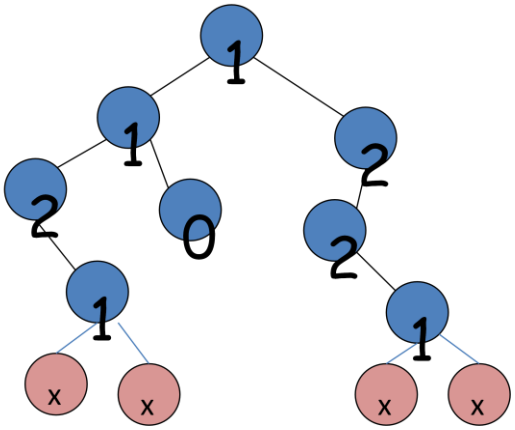
Compitino di Programmazione del 12 marzo 2014

Si tratta di aggiungere nodi ad un albero binario e di farlo per livelli. Facciamo subito un esempio:





Supponiamo di voler aggiungere nodi al livello 3 dell'albero a sinistra. Potremmo aggiungere al massimo 4 nodi perché l'albero "offre" solo 4 posti per nodi a livello 3. Se volessimo aggiungere nodi al livello 4 (ancora dell'albero a sinistra), potremmo di nuovo aggiungere solo 4 nodi ottenendo l'albero seguente:



Immaginiamo di avere a disposizione 10 nodi da aggiungere al livello 4. Visto che ne possiamo inserire solo 4, dovremo restituire il valore 6 per poter inserire i 6 nodi ancora disponibili in qualche altro modo. Per esempio a livello 5. Nel caso che provassimo ad inserire nodi a livello 5 direttamente nel primo albero (in alto a sinistra), non potremmo inserire alcun nodo visto che l'albero non ha nodi a livello 4. A maggior ragione se provassimo ad inserire nodi a livello 6, 7 ecc. L'albero "offre" posti per inserire nuovi nodi al livello 2 (1 posto), a livello 3 (4 posti) e al livello 4 (4 posti) e basta. Ovviamente potremmo aggiungere un numero arbitrario di nodi all'albero di partenza, aggiungendo nodi al livello 2, poi al livello 3, poi al livello 4, poi al 5, e così via. Nel seguito con Offre(r,lev) indichiamo il massimo numero di nodi che si può aggiungere a livello lev nell'albero r.

Si chiede di scrivere 2 funzioni ricorsive come segue. La funzione int level(nodo*&r, int dim, int lev, ifstream & INP) deve soddisfare le seguenti pre- e post-condizioni:

PRE_level=(r è albero corretto, dim >=0, lev>=0, r=vr, INP contiene almeno dim interi)

POST_level=(r è ottenuto da vr aggiungendo a vr min(Offre(vr,lev), dim) nodi a livello lev da sinistra a destra, la funzione deve restituire col return dim-Offre(vr,lev) qualora questa quantità fosse positiva e altrimenti deve restituire 0; i nodi (eventualmente) aggiunti a vr devono ricevere il loro campo info da letture effettuate su

La seconda funzione da realizzare è nodo* build(nodo*r, int dim, int lev, ifstream & INP) e deve soddisfare le seguenti pre- e post- condizioni:

PRE_build=(r albero corretto, dim>=0, lev>=0, INP contiene almeno dim interi, r=vr, dim=vdim)

POST_build=(r è ottenuto da vr aggiungendo a vr dim nuovi nodi (con campo info letto da INP e tali che questi nodi siano inseriti a partire dal primo livello l tale che Offre(vr,l) >0 , poi al livello l+1, l+2 e così via, fino ad esaurire i dim valori che sono su INP; per ogni livello i nodi vanno inseriti da sinistra a destra)

Correttezza: dimostrare in modo induttivo la correttezza di level.

Viene fornito un main che esegue prima level che cerca di inserire dim0 nuovi nodi al livello 2 in un albero X costruito da programma e successivamente stampa su "output" l'albero ottenuto in forma lineare e stampa anche il numero di nodi che resterebbero da inserire, qualora X non permettesse di inserire tutti i dim0 nodi al livello 2. Si osservi che il main legge da INP i valori eventualmente non inseriti in X. Successivamente il main invoca build due volte, prima su X aggiungendo ad X dim1 nodi e poi partendo dall'albero vuoto e aggiungendo dim2 nodi. In entrambi i casi l'albero ottenuto viene stampato su "output" in forma lineare. La funzione di stampa lineare e di costruzione di X sono date.