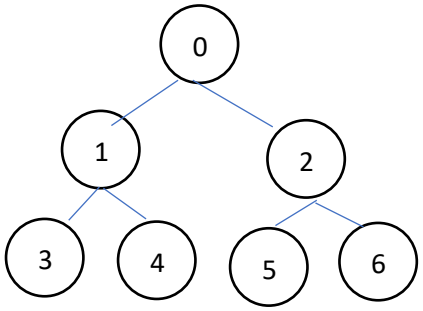
Il progetto consiste nella realizzazione di un buddy allocator utilizzando come struttura ausiliaria una bitmap.
Il concetto generale è analogo a quello di un classico buddy allocator, quindi avremo una allocatore di blocchi a dimensione variabile, ma in questo caso, per memorizzare i blocchi liberi e quelli occupati, viene utilizzata unicamente una bitmap.

Tale bitmap serve per rappresentare l'albero della memoria nel quale ogni bit corrisponde a un nodo e ogni livello dell'albero corrisponde a un livello del buddy allocator.



BITMAP

0	0	0	0	0	0	0
0	1	2	3	4	5	6

La bitmap è rappresentabile tramite un array inizialmente inizializzato a 0:

- ➔ 0 = posizione libera
- ➔ 1 = posizione occupata/ parzialmente occupata (il nodo è stato diviso in 2)

Le operazioni principali implementate per l'utente sono 2:

1. MALLOC:
    a. Data la dimensione richiesta da allocare (si aggiunge a questa anche la memoria necessaria per l'indice), si calcola il livello massimo a cui questa potrebbe essere allocata.
    b. Si verifica se esiste un blocco libero corrispondente a quel livello
    c. Se esiste, si assegna tale valore ad una variabile idx, si salva idx nei primi 4 byte della memoria da allocare e si restituisce all'utente un puntatore alla memoria + 4.
    d. Se invece idx=-1, quindi non è valido si ritorna errore.
2. FREE:
    a. Si leggono i primi 4 byte per trovare l'indice della cella di memoria da liberare
    b. Si verifica che il valore sia valido<2^num_bit
    c. Si setta il bit idx della bitmap a 0, cioè lo si libera, e si fa la stessa cosa.
    d. In un ciclo while si verifica se anche il suo "buddy" è libero e in tal caso si fa il merge di idx e del suo buddy, liberando quindi anche il padre. Si uscirà dal ciclo solo una volta trovato un buddy allocato o dopo essere arrivato alla radice.