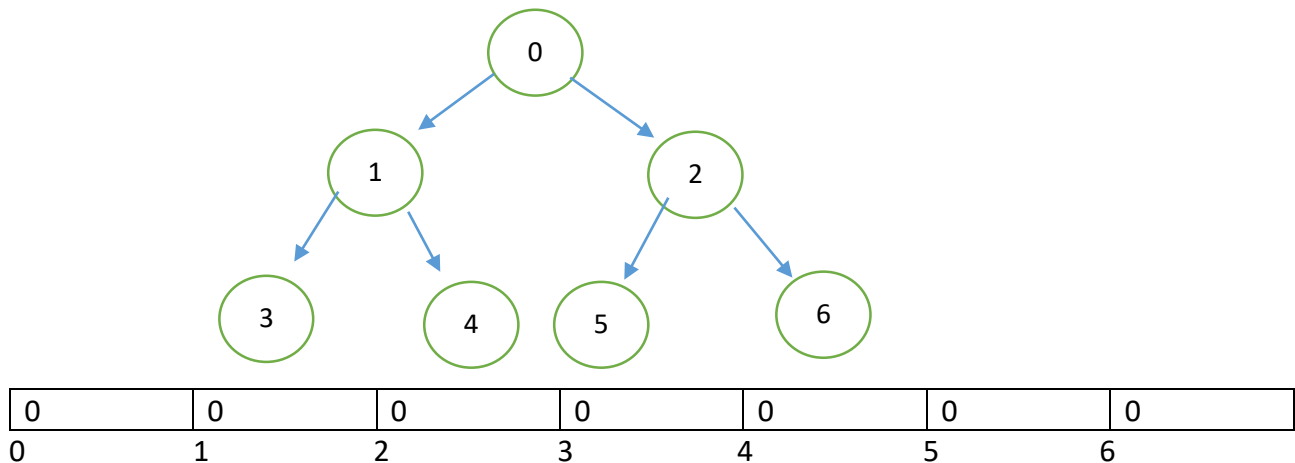


Il progetto consiste nel realizzare un buddy allocator utilizzando come struttura ausiliaria una bitmap.

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

Questa 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



La bitmap è rappresentabile tramite un array inizialmente inizializzato a 0, dove lo 0 rappresenta una posizione libera, mentre l'1 rappresenta una posizione occupata o parzialmente occupata.

Le operazioni implementate per l'utente sono 2:

1. MALLOC:

- Data la dimensione richiesta da allocare (aggiungendo anche la memoria necessaria per l'indice), si calcola il livello massimo a cui questa potrebbe essere allocata
- Si verifica se esiste un blocco libero corrispondente a quel livello
- Se esiste, si assegna tale valore ad una variabile `idx`, si salva `idx` nei primi 4 byte della memoria e si restituisce all'utente un puntatore a memoria `+4`.
- Se invece `idx=-1`, ritorna un errore dato che l'indice non è valido

2. FREE:

- Si leggono i primi 4 byte per trovare l'indice della cella di memoria da liberare
- Si verifica che il valore sia valido, cioè minore di $2^{\text{num_bit}}$
- Si imposta il bit `idx` della bitmap a 0, liberandolo
- In un ciclo si verifica se anche il suo "buddy" è libero. In tal caso, si fa merda di `idx` e del suo buddy, liberando anche il padre. Si uscirà dal ciclo solo una volta trovato un buddy allocato o dopo essere arrivato alla radice