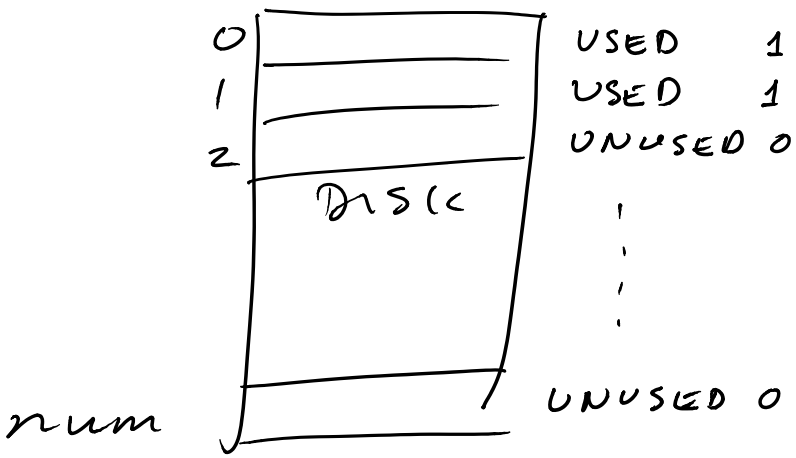
# SAT - How to represent?





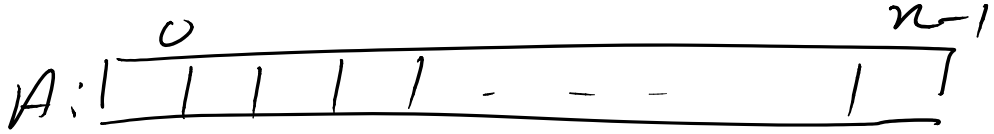
⌈10000/1024⌉ ← 128 bytes * 8 bits per byte

⌈9.765 tracks⌉ = 10 tracks

---

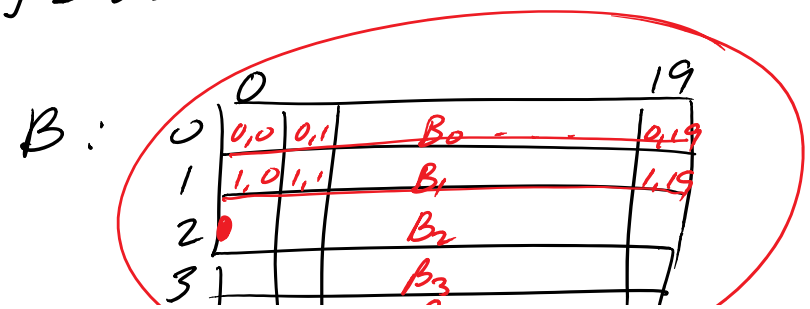
ASIDE

n = 100

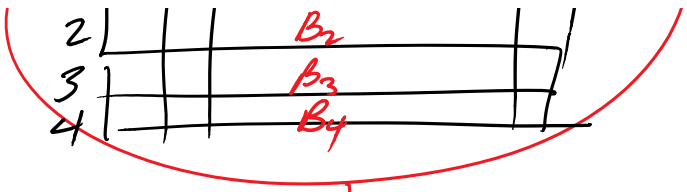



int A[n];

A[i] = addr(A) + i * (size of int)




is there 20 space in block 0




2D space in physical memory?

int B[5][20]; C++ row major order

physical


$B[i][j] = addr(B) + i * (\text{row size}) + j$

$= addr(B) + i * (ub_2 - lb_2 + 1) + (j - lb_2)$

$lb_1 .. ub_1$ $lb_2 .. ub_2$
$0 .. ub_1$ $0 .. ub_2$

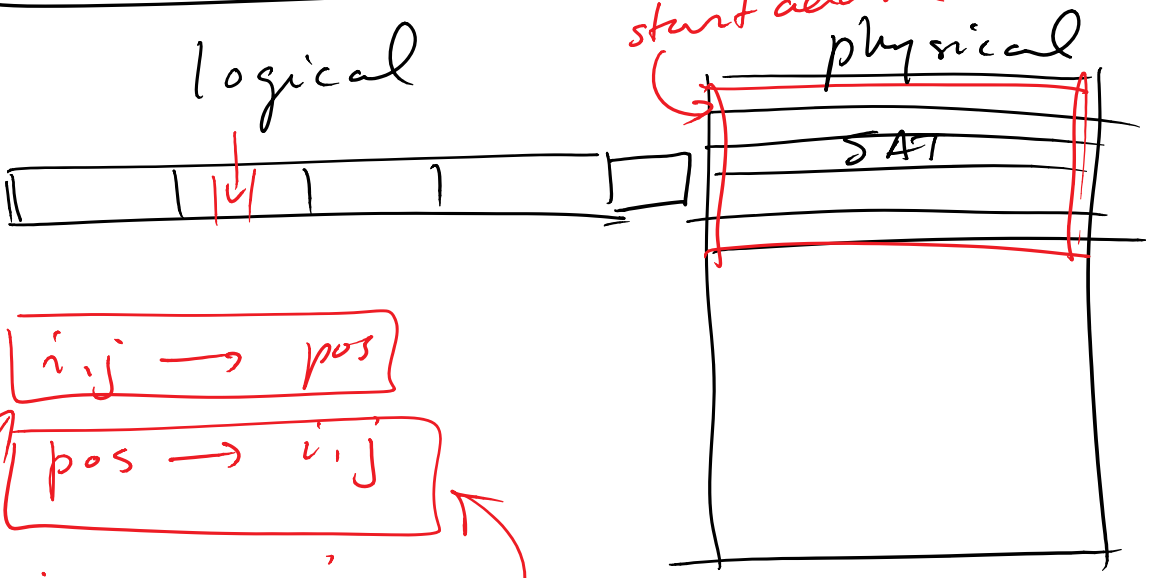
$= addr(B) + i * (19 - 0 + 1) + (j - 0)$

$= addr(B) + (i * 20 + j) * \text{elt size}$

$B[2][0] = addr(B) + 2 * 20 + 0$

$= addr(B) + 40$

start addr (start track)



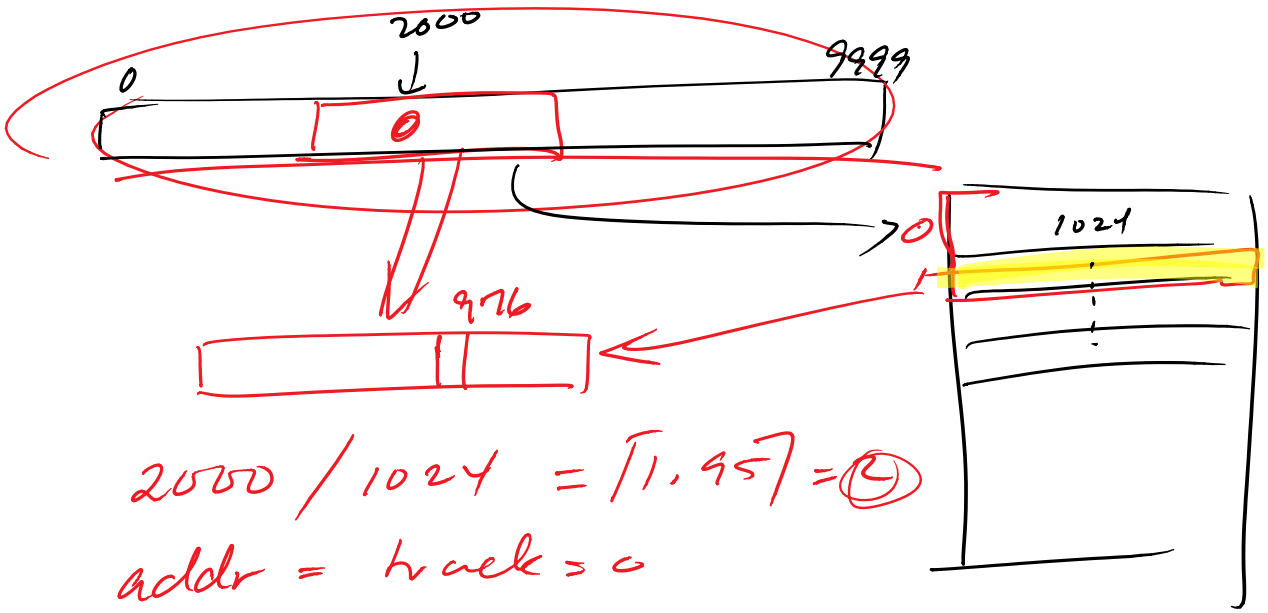

i, j → pos

pos → i, j

i * sz + j

i = pos / sz

j = pos % sz




2000 / 1024 = ⌈1.95⌉ = 2

addr = track = 0

= track = 1

2000 % 1024 = 976

---

#include <bitset>

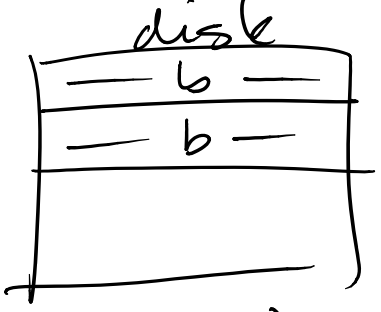
bytes (1 char)

8 bits

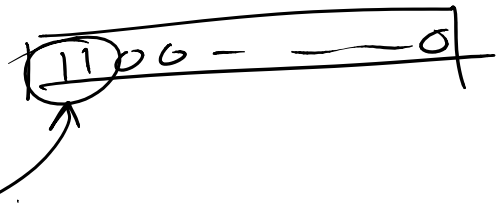
int main() {

bitset<1024> b;

WriteDisk((unsigned char*)&b, 0);

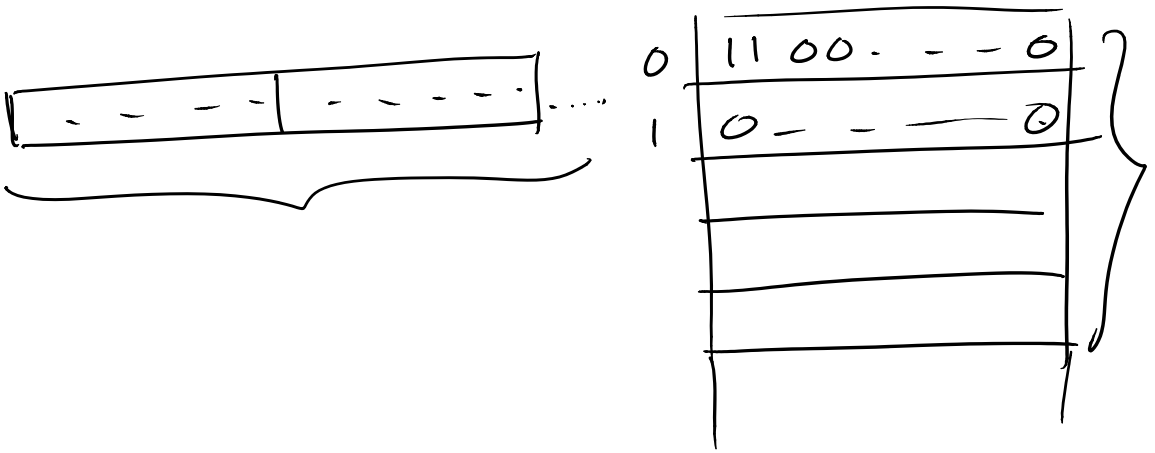



WriteDisk((unsigned char*)&b, 1);




b[0] = 1;
b[1] = 1;

WriteDisk((unsigned char*)&b, 0);



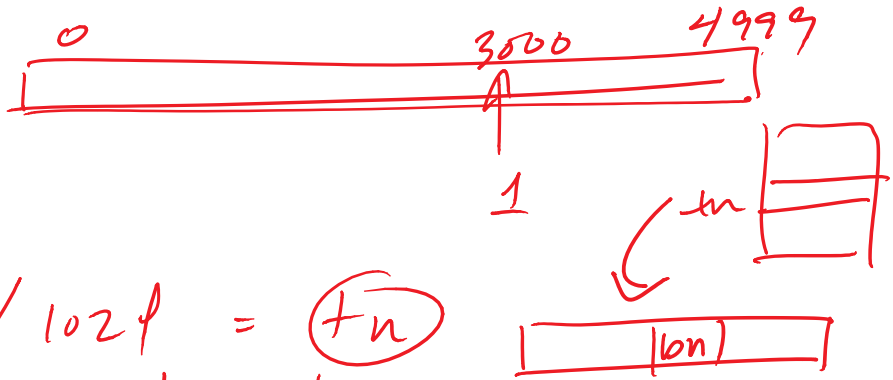

---

```
class SAT {
public:
    SAT(size);
    void setBit(int pos);
```

5000

3000

```
void setBit(int pos);
void resetBit(int pos);
bool testBit(int pos);
};
```




pos / 1024 = tn

pos % 1024 = bn