113

$$\left.\begin{array}{l} 1\,000\,000\,000\,000\,0\,666 \\ \underbrace{0000000000000}_{13}\,1 \end{array}\right\}\text{prime}$$

Dynamic programming = 1) clever recursion
2) clever implement

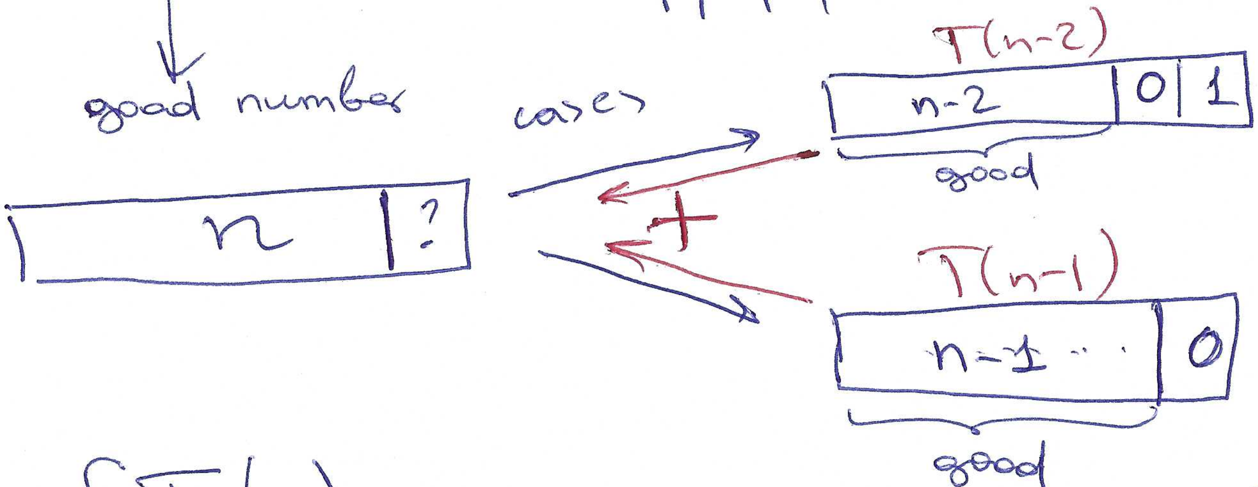
MAT 258

Count # of n-bit numbers that ~~11~~:

0000
0101 } ✓
1001

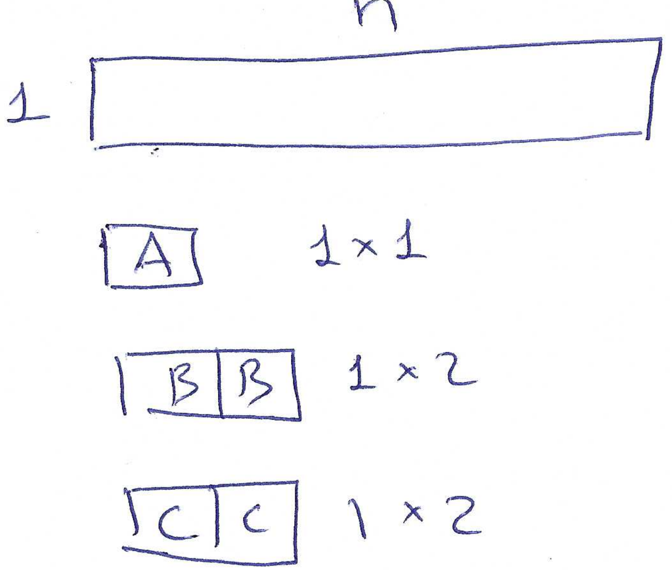
1100 —

1111 —



$$\left.\begin{cases} T(n) = T(n-2) + T(n-1) \\ T(1) = 2 \\ T(2) = 3 \end{cases}\right\} \text{Fib.}$$

tile



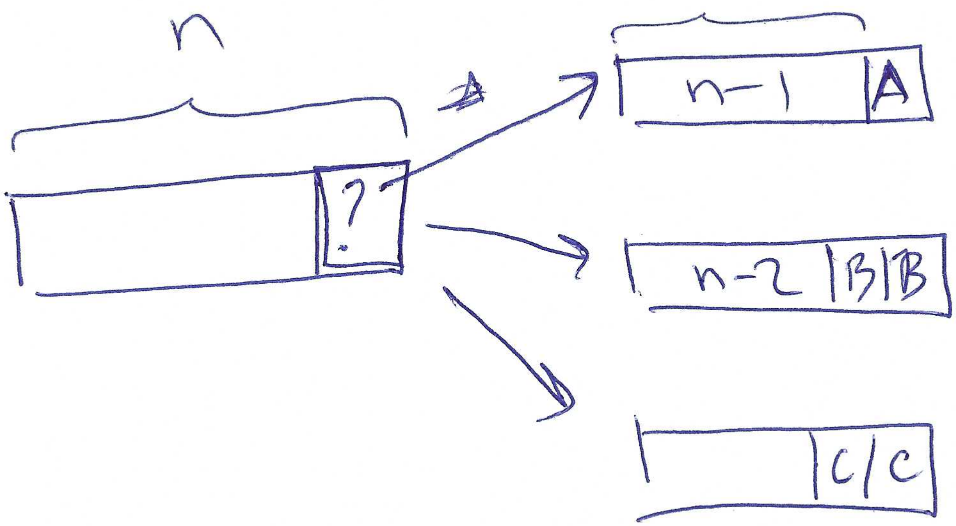
tilings

$n = 4$

AAAA
AABB
AACC
BBAA
CCAA

ABBA
ACCA

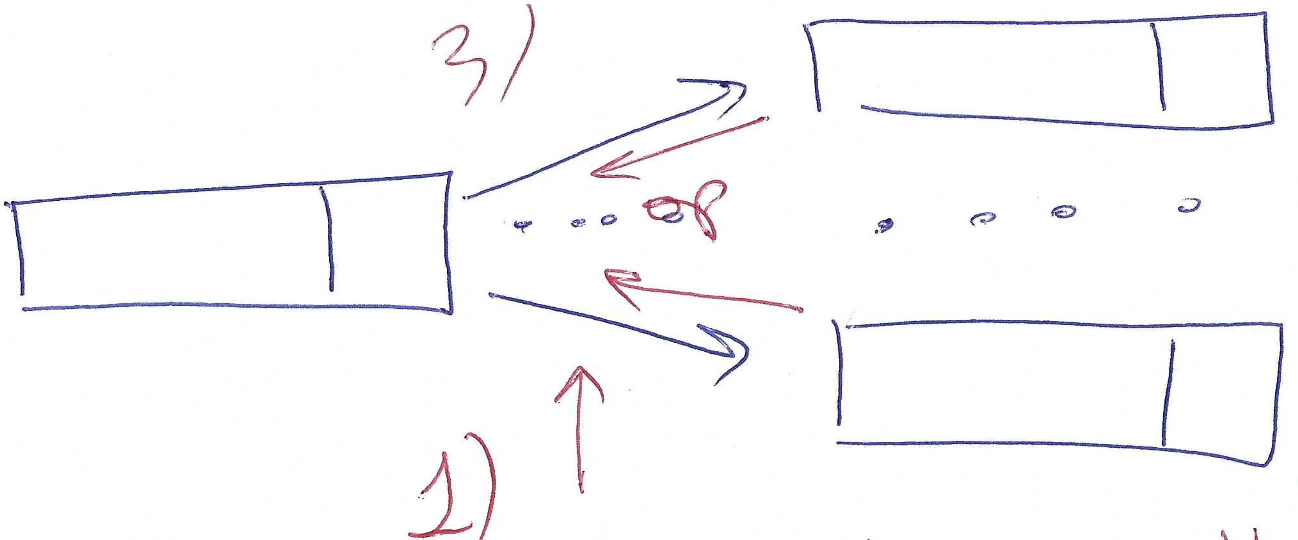
BBBB
CCCC
BBCC
CCBB



$T(3) = 3 + 2 \cdot 1 = 5$

$T(4) = 5 + 2 \cdot 3 = 11$

$$\begin{cases} T(n) = T(n-1) + T(n-2) + T(n-2) \\ T(1) = 1, \quad T(2) = 3 \end{cases}$$



add info
2) rec.call

cases: 1) cover all
2) cases are not overlapping

Plan for clever recursion.

```
FR(n) {
  if( n<2) return n;
  r = FR(n-2) + FR(n-1);
  return r;
}
```

rec

$K \to K_{-2}$ diff $K_{-1} \to K$

duplicates!!!!

remove 0's

AVL - worst case!! $\Rightarrow$ $\Rightarrow$ exponential

$\Rightarrow$ fewest number of nodes φ^n

$\varphi = \frac{1+\sqrt{5}}{2}$

```
FI(n) {
  t[n+1]; // last index is n
  t[0]=0;
  t[1]=1;
  for(i=2; i<n; ++i) {
    t[i]=t[i-2]+t[i-1];
  }
  return t[n];
```

linear

- 6 -

LCS

g f c e a b d h

f a d e c b

fcb is CS

fce is not

Longest

g f c e a b d h

f a d e c b

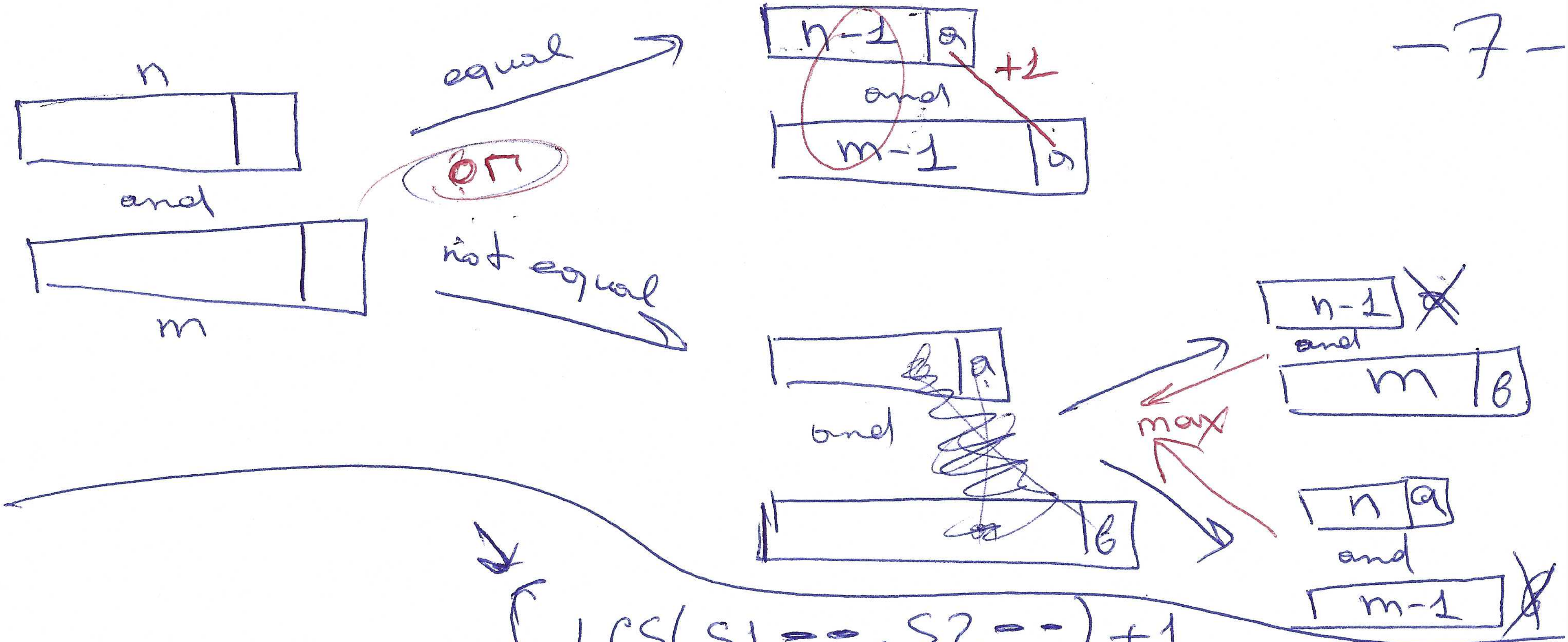
subsequence (str1) "c a h" order

subarray [....]

subset = subseq no order

de is not s/s in str1

is s/s in str2



$$LCS(S1,S2)=\begin{cases} LCS(S1--,\ S2--)+1, & S1.back()==S2.back() \\ max(LCS(S1--,\ S2),\ LCS(S1,\ S2--)), & \text{otherwise} \\ 0 & \text{if } S1.size()==0 \text{ or } S2.size()==0 \end{cases}$$