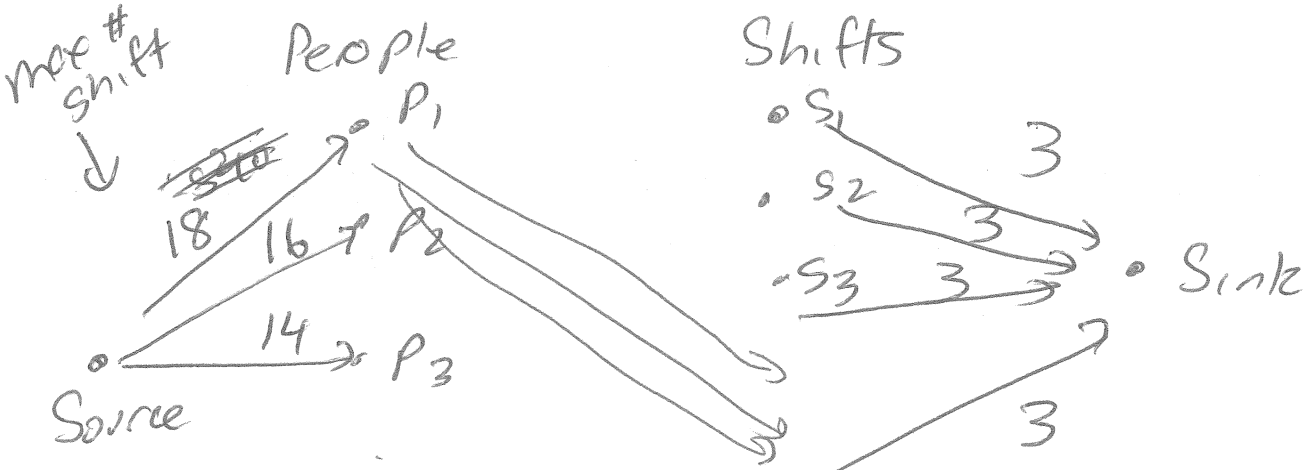
COP 3503 10/29/26

1. Museum Cards
2. Chess Competition
3. Integer Multiplication if time

→ Orisinal Q: What's the maximum # guards I can guarantee ~~for~~ for all shifts?

Simplification: Can we cover all 48 shifts with k guards?



Run Max Flow
if flow == 3*48
then 3 is possible
otherwise it's not
(real ans < 3)

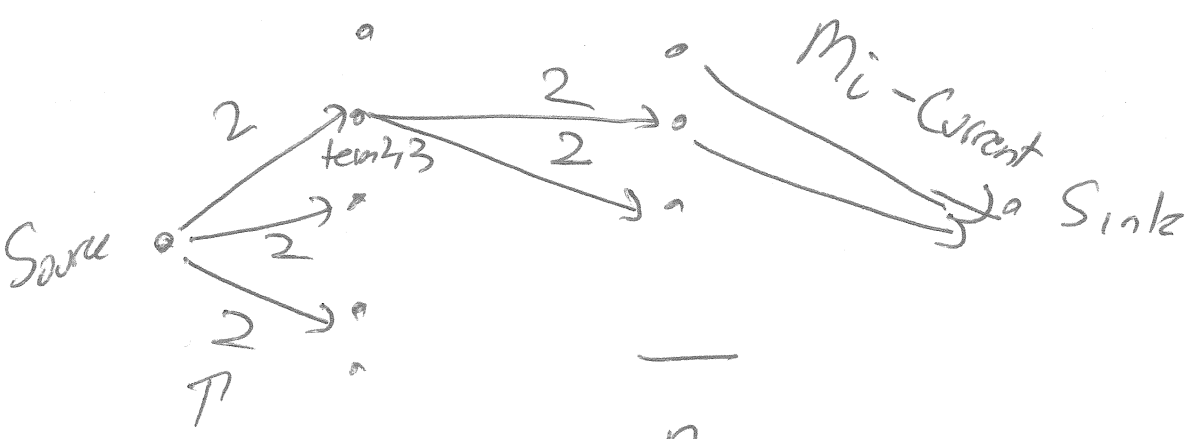
8-10 (shifts 16-19)
9-12 (shifts 20-23) adj
18-20 (shifts 36-39)

Given current standings
list of unplayed games
can team i win? (tie for most pts

(1) Assume team i wins out
compute max score (M_i)

(2) If another team has more → return false.

(3) Set up Network Flow Graph.
Idea → can all other teams score $\leq M_i$?



T
k games left

n
teams

n^2 vertices
(i,j)
$i<j$

If flow = 2k
then team i can win else it can't!

Divide + Conquer

Fancy way to say "recursion"

Polynomial / Integer Multiplication

$$(3x^4 + 2x^2 + x - 6) \cdot (7x^5 - 3x^4 + 2x^3 + x^2)$$

	0	1	2	3	4
poly1	-6	1	2	0	3
poly2	0	0	1	2	-3	7

```
for (int i=0; i<poly1.length; i++)
    for (int j=0; j<poly2.length; j++)
        ans[i+j] += poly1[i]*poly2[j];
```

If both degree n polynomials
run-time $O(n^2)$.

Simplify and assume our polynomials are of degree $2^k - 1$, stored using 2^k coefficients.

$$f_1(x) = h_1(x) \cdot x^{2^{k-1}} + l_1(x)$$
$$f_2(x) = h_2(x) \cdot x^{2^{k-1}} + l_2(x)$$

$$\longrightarrow h_1(x)h_2(x) \cdot x^{2^k} + h_1(x)l_2(x) \cdot x^{2^{k-1}} + h_2(x)l_1(x) \cdot x^{2^{k-1}} + l_1(x)l_2(x)$$

$$(x^3 + 2x^2 - x + 7)(x^3 - x^2 + 3x + 2)$$

$$f_1(x) = (x+2)\cdot x^2 + (-x+7)$$
$$f_2(x) = (x-1)\cdot x^2 + (3x+2)$$

Let $T(n)$ represent run-time of recursive mult alg where $n = 2^k - 1$.

$$T(n) = 4 \times T\left(\frac{n}{2}\right) + O(n)$$

$A = 4 \quad B = 2 \quad k = 1$

run-time $\log_2 4 > 1$ so

run-time is $O(n^{\log_2 4}) = O(n^2)$

$$f_1(x) f_2(x) = h_1(x)h_2(x)\cdot x^{2k} + \left(h_1(x)\ell_2(x) + h_2(x)\ell_1(x)\right)\cdot x^{2^{n-1}} + \ell_1(x)\ell_2(x)$$

Consider Product $(h_1(x) + \ell_1(x))(h_2(x) + \ell_2(x))$

COMPUTE $\rightarrow P_3 = h_1(x)h_2(x) + h_1(x)\ell_2(x) + \ell_1(x)h_2(x) + \ell_1(x)\ell_2(x)$

COMPUTE $h_1(x)h_2(x)$ P_1

COMPUTE $\ell_1(x)\ell_2(x)$ P_2

$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$

$$\text{MIDTERM} = P_3 - P_1 - P_2 \rightarrow O(n^{\log_2 3})$$

STRASSEN'S ALGORITHM