(7) $Factors = \{\langle n \rangle \in \mathbb{Z} \mid \exists_{x_1 \dots x_n} : x_1^{a_1} * \dots * x_n^{a_n} = n \}$ $\rightarrow x_1 \dots x_n \leq n$

$Factorization = \{\langle n \rangle \mid \exists_p : 1 < p < n \wedge n \% p = 0\}$

$= -D = \{\langle n, x \rangle \mid \exists_p : 1 < p < x \wedge n \% p = 0\}$

$\downarrow$

$1 \leq x \leq n$

a) Factorization-D is NP and co-NP

Verificador NP: donat n i x

Verificador co-NP:

```
input n, x {
    Si n%x = 0
        return TRUE
    else
        return FALSE
}
```

$\Rightarrow$ igual amb $\begin{cases} TRUE \\ FALSE \end{cases}$

b) Si P=NP, demostra que Factorization $\in FP$ (computables a temps polinòmic)

Factorization $\in NP$, si provem que també és NP-hard, llavors com P=NP, Factorization és FP, ja que podem reduir en temps P a l'algoritme que resol en temps P.

Factorization $\leq_n^p$ SAT:

$n = \lfloor \sqrt{N} \rfloor$

Factorization $\longrightarrow$ SAT

$n = N$

$N \in \mathbb{Z}$: $F = \bigwedge_{i=1}^{\binom{n}{2}} C_i$ $\quad b_j = \{x_2 \dots x_n\}$

$$C_i = \begin{cases} x_a \vee x_b & \text{si } a*b = N \\ \neg x_a \vee \neg x_b & \text{si } a*b \neq N \end{cases} \quad a, b \in [2, n]$$

output: p or PRIME : output: $x_{min} = 1$ or UNSAT

Cost: $O(\binom{n}{2}) \simeq O(n^2)$, que és P

Correctesa: No sé :(com demostrar-ho

c) Si P=NP, FACTORS ∈ FP

Raonament similar a apartat b), si volem factors a SAT, que és NP-complet, llavors com P=NP, existeix un algoritme polinòmic que redueix NP a P, per tant, redueix ~~SAT~~ de Factors a FP.

Factors ⟶ SAT

input: $N \in \mathbb{N}$: $F = \bigwedge_{i=1} C_i$ $C_i = \{$

$L = \{ x_1 \dots x_n \}$

Primes

(Primes is in P)

$n \in [N]$

Fins que $a^b > N$

Igual que apartat anterior, però amb exponents ($X_{2\_1}, X_{2\_2} \dots X_{n\_exp}$)

Un cop resolt el SAT, l'output és una llista de tots els literals = 1 amb l'exponent més gran