Ý tưởng

Ta nhận xét từ việc là:
- Với một thời gian T cố định thì chúng ta có thể tính được tổng lượng sản phẩm hoàn thành của một phân xưởng.
- Nếu mà T càng tăng thì số lượng ấy cũng càng tăng.

Vậy ta sẽ chia T_1, T_2 là thời gian cho phân xưởng 1 và phân xưởng 2.

Dễ thấy là chúng ta phải thỏa $T_1 + T_2 \leq T$ nhưng mà để phân xưởng 1 và phân xưởng 2 tận dụng hết thời gian hết mức có thể (để tạo ra nhiều sản lượng nhất có thể) thì chúng ta sẽ cho nó là $T_1 + T_2 = T$.

Như vậy chúng ta sẽ tiến hành tìm kiếm nhị phân (TKNP) cho T_1 như sau.

- Với T_1 chúng ta sẽ tạo được S_1 là lượng sản phẩm của phân xưởng 1 làm được trong thời gian T_1. Tương tự thì T_2 sẽ có S_2, thì sản phẩm hoàn thành được cuối cùng chính là $\min(S_1, S_2)$.
- Nếu $S_1 > S_2$ đồng nghĩa là ta đang cho thời gian T_1 tạo ra quá nhiều sản phẩm so với phân xưởng 2, do vậy ta sẽ giảm T_1 để tăng T_2 lên.
- Tương tự nếu $S_1 < S_2$ đồng nghĩa ta đang cho thời gian phân xưởng 1 quá ít để tạo ra sản phẩm, do vậy ta sẽ tăng T_1 lên để giảm T_2 xuống.

Áp dụng TKNP trên đoạn $[0, T]$ cho T_1 xong ta sẽ có được đáp án chính là $\min(S_1, S_2)$.

Độ phức tạp thời gian

$O((N + M) \times \log(T))$ với N, M lần lượt là độ dài của mảng 1 và 2.

Notes

Ngoài ra chúng ta cũng nên lưu ý là số lượng sản phẩm làm được ở mỗi phân xưởng có thể lên đến $N \times T$ dẫn đến là tràn số nếu mà lưu ở kiểu dữ liệu int.