1 Lista 6, Zadanie 6

1.1 Wstęp agitacyjny

Zrozumienie treści zadania 6 z listy 6 (być może przez mnogość 6) stanowi dla mnie pewien problem. Na przykład jest w nim napisane, że mozemy zamówić dowolną liczbę paczek. Jeżeli możemy zamówic ich dowlona liczbę to każdego dnia wieczorem wystarczy zamówić tyle paczek ile sprzedamy następnego dnia (zakłądając, że przychodzą rano). Jeżeli paczki muszą być składowane przed sprzedażą, to zadanie rónwież nie ma sensu bo, nie sprzedamy ich więcej niż W, więc ponownie wystarczy ich zamówić tyle ile na ten dzień potrzeba. W obliczu tylu niejasności chciałbym zaproponować interpretacje zadania, mam nadzieję, ze nie pozbawi go ona animuszu i werwy oraz nie spłyci. Algorytm, kóry zaproponuję też jest dość ciekawy. Jeżeli zmyślam to bardzo przepraszam, mam nadzieję, że to rozwiązanie też będzie w jakiś sposób rozwijające.

1.2 Interpretacja

Podejrzewam, że zadanie miało na celu opracowanie algorytmu minimalizujacego koszt przetrzymywania materiału w spichlerzu, z zasadą, że nadmiar którego nie mozemy schować do spichlerza psuje się od razu, natomiast ten pozsostając w spichlerzu jest zakonserwowany i nieśmiertelny. Dostawa odbywa się rano, przez co możemy operować tym co mieliściemy w spichlerzu z wczoraj oraz tym co zamówiliśmy. Koszt c płacimy tylko za rzeczy pozostawione w spichlerzu, no i jesteśmy ekologiczni co oznacza, że zamawiamy tyko tyle ile potrzebujemy, bez nadmiaru (dziwny zapis "Zamówienie dostawy dowolnej liczby paczek kosztuje P", jedna paczka i sto paczek kosztuje P?). Musimy więc dać sobie jakieś rozsądne ograniczenia:

1. W - pojemność naszej stacji przeładunkowej.
2. c - koszt składowania jednej jednostki przez *noc*.
3. P - niech koszt zamówienia będzie stały, ale ilość paczek ograniczona np. do W.
4. T - wektor zakupów $[0, t_1, ..., t_n]$, o długosci $n + 1$. t_i to przewidziana ilosć zakupionych od nas jednostek przypadająca na i-ty dzień. Dzień 0 to dzień startowy, w którym dochodzi tylko do zamówienia, bez sprzedaży. Założenia:
   (a) Musimy spełnić t_i dla każdego i. (Rozsądne?).
   (b) Aby poprzednie założenie było prawdziwe musimy, nałożyć ograniczenie na zamówienie, niech będzie to $2W$ (ogólnie to $W+ograniczenie(P)$). W przeciwnym razie biorąc 1. i 2. niemożlwie byłoby spełnienie takiego zamówienia.

Także, postaram się rozwiązać tak postawione zadanie.

1.3 Algorytm

Stworzymy wektor zamówień $Z = [z_0, z_1, ..., z_n]$. Algorytm będzie rozważał dwa przypadki: taki, w którym wystarczy nam zamówienie na następny dzień i taki w którym rzeczywiście będziemy potrzebowali spichlerza.
Czyli: dla dnia i, kiedy nie wystarczy dostawa z rana cofamy się po Z i znajdjemy pierwszy, kolejny dzień kiedy damy radę zamówić więcej i tak aż spełnimy zamówienie.

```
for i = 0 to n do
    if T[i+1] ≤ W then
        Z[i] = T[i+1]
    else
        for j = i to j = 0 do
            if W - Z[j] < T[i+1] then
                Z[j] = W
                T[i+1] = T[i+1] - (W - Z[j])
            else
                Z[j] = Z[j] + T[i+1]
            end
        end
    end
end
```

Algorithm 1: Puerto Rico Algorithm