Zadanie 8

Najprostszym podejściem byłoby nie przechowywać żadnej dodatkowej informacji, tylko przy każdym zapytaniu $sum(i, j)$ w locie wyliczać $A[i] + A[i + 1] \ldots + A[j]$. Oczywiście w takim przypadku nie mamy tablicy B, zatem nie używamy żadnej dodatkowej pamieci. Ale koszt pojedynczego zapytania może być duży – wyliczenie $sum(i, j)$ ma każdorazowo koszt $O(N)$ (bo j – i może być rzedu $O(N)$, np. $(j = N, i = \frac{N}{3})$.

Inne podejście to użycie dodatkowej pamieci, które pozwoli na zmniejszenie kosztu pojedynczych zapytań.
Pomyślmy o nastepujacym podejściu – w tablicy $B[0 \ldots N]$, w komórce $B[i]$ przechowujemy sume elementów tablicy A od pierwszego do i-tego włacznie. Tj. $B[i] = A[1] + A[2] + \ldots + A[i], B[0] = 0$. Wtedy oczywiście

$$sum(i, j) = B[j] – B[i - 1] = A[1] + \ldots + A[i - 1] + A[i] +$$
$$+ \ldots \text{ A[j] – (A[1] + \ldots A[i-1]) = A[i] +.. + A[j].}$$

Czyli kosztem użycia $N + 1$ komórek pamieci (dodatkowa pamieć $\Theta(N)$) możemy wyznaczyć $sum(i, j)$ w czasie stałym – mamy odczytanie 2 komórek pamieci odpowiadajacych $B[i]$ i $B[j]$ oraz jedno odejmowanie – zakładamy, że koszt tych atomowych operacji jest stały.

Pozostaje tylko odpowiedzieć, ile czasu zajmuje wyliczenie tablicy B. Możemy to zrobić w czasie $\Theta(N)$ przy pomocy takiej prostej procedury

```
B[0] = 0

FOR i FROM 1 TO N DO

   B[i] = B[i-1] + A[i]

ENDFOR
```

Podsumowujac – nasz algorytm nie używa dodatkowej pamieci i nie wykonuje żadnych obliczeń wstepnych, ale koszt pojedynczej odpowiedzi na pytanie o sume jest rzedu $O(N)$, albo potrzebujemy pamieci rzedu $\Theta(N)$, mamy wstepna faze obliczeń trwajaca $\Theta(N)$, ale udzielenie odpowiedzi na każde kolejne pytanie postaci $sum(i, j)$ zajmuje czas stały $\Theta(1)$.