Zadanie 4 (25%)
Wyjeżdżasz na długą wycieczkę do Doha, rozpoczynając ją od budynku głównego PWr. Po drodze znajdziesz n hoteli, i-ty w odległości k_i od PWr. Dodatkowo wiesz, ze $k_1 < k_2 < \ldots < k_{n-1} < k_n$ i n-ty hotel znajduje się już w Doha. Po drodze możesz się zatrzymać tylko w wyznaczonych hotelach, ale możesz sobie wybrać w których z nich. Oczywiście musisz się zatrzymać w n-tym hotelu, który jest Twoim celem. Chciałbyś przemierzać 543 km dziennie, ale to może być niewykonalne z powodu odległości między hotelami. Załóżmy, że pokonując x km danego dnia musisz zapłacić $\frac{(543-x)^2}{x}$ za ten dzień. Chcesz zaplanować swoją podróż tak aby zminimalizować sumę opłat za wszystkie dni. Podaj algorytm (o złożoności co najwyżej wielomianowej w zależności od n), który wyznaczy optymalny ciąg hoteli, w których powinieneś się zatrzymać podczas podróży. Udowodnij poprawność i złożoność obliczeniową algorytmu.

Rozwiązanie: W celu znalezienia rozwiązania tego zadania posłużymy się metodologią programowania dynamicznego, czyli zaczniemy od zdefiniowania sobie zbioru podproblemów, kolejności na tych podproblemach oraz relacji pomiędzy tymi podproblemami.

Najpierw skupimy się na znalezieniu minimalnego kosztu wycieczki, a potem odtworzymy postoje.

Podproblem zdefiniujemy jako: $L(i)$ - minimalny koszt dojechania do i-tego hotelu.

Kolejność podproblemów jest naturalnie zadana przez to, że hotele są na kilometrach $k_1 < k_2 < \ldots < k_{n-1} < k_n$, czyli podproblemami $L(i)$ są wszystkie $L(j)$ dla $j < i$. Zauważmy, że PWr możemy zdefiniować jako początek trasy czyli "hotel 0"i założyć, że dotarcie do niego nic nie kosztuje $L(0) = 0$.

Zdefiniujmy teraz relację pomiędzy podproblemami. Relacja ta wynika wprost ze wzoru na koszt pokonywania x kilometrów dziennie:

$$L(i) = \min_{0 \le j < i} \left\{ L(j) + \frac{(543 - (k_i - k_j))^2}{k_i - k_j} \right\}.$$

Podproblemy będziemy rozwiązywać w zdefiniowanej kolejności zapamiętując ich rozwiązania.

Pseudokod:

$L(0) = 0$ ▷ inicjalizacja zmiennych
$prev(0) = 0$ ▷ zmienna $prev$ będzie potrzebna do odtworzenia optymalnej trasy
for $i = 1$ to n **do**
  $L(i) = \infty$
  $prev(0) = null$
end for
for $i = 1$ to n **do**
  $L(i) = \min_{0 \le j < i} \left\{ L(j) + \frac{(543-(k_i-k_j))^2}{k_i-k_j} \right\}$
  $prev(i) = \arg\min_{0 \le j < i} \left\{ L(j) + \frac{(543-(k_i-k_j))^2}{k_i-k_j} \right\}$
end for

Łatwo zauważyć, że zdefiniowana relacja na $L(i)$ sprawdza wszystkie „sensowne” możliwości wyznaczenia optymalnej drogi do Doha, gdzie „sensowne” oznacza, że rozwiązując i-ty Podproblem sprawdzamy rozwiązania mniejszych podproblemów (dojazd do j'tego hotelu) i koszt drogi od j-tego hotelu do i-tego. Relacja jest również zdefiniowana w taki sposób, aby wybrać optymalne rozwiązanie, bo wyszukuje rozwiązanie o minimalnym koszcie z wszystkich „sensownych" możliwości. Można też zauważyć, że tak naprawdę sprowadziliśmy nasz problem do problemu znalezienia najkrótszej ścieżki w DAG'u, gdzie waga krawędzi pomiędzy j-tym a i-tym wierzchołkiem to $\frac{(543-(k_i-k_j))^2}{k_i-k_j}$.

Złożoność: wykonujemy dwie pętle długości n. W pierwszej pętli wykonują się operacje o złożoności $\Theta(1)$, wiec złożoność pierwszej pętli wynosi $O(n)$. W drugiej pętli, operacja znalezienia minimum (w implementacji $\arg\min$ dostajemy wraz ze znalezieniem minimum) dla i tego przebiegu ma złożoność $O(i)$. Zatem złożoność drugiej pętli to:

$$\sum_{i=1}^{n} O(i) = O\left(\sum_{i=1}^{n} i\right) = O(n^2).$$

Zatem złożoność całego algorytmu to $O(n^2)$.