2017/2018, kolokwium 1, zadanie 1

a) * oznaczmy elementy przez a_1, a_2, a_3, a_4 (w tej kolejności występują w ciągu)

* jeśli ciąg ma co najwyżej jedno ekstremum, to jego elementy muszą być powiązane którąś z poniższych relacji:

(1) $a_1 < a_2 < a_3 < a_4$

(2) $a_1 > a_2 > a_3 > a_4$

(3) $a_1 < a_2 > a_3 > a_4$

(4) $a_1 < a_2 < a_3 > a_4$

(5) $a_1 > a_2 < a_3 < a_4$

(6) $a_1 > a_2 > a_3 < a_4$

* zliczamy możliwe permutacje a_1, a_2, a_3, a_4, które pasują do powyższych relacji

* przypadek (1) dokładnie wyznacza permutację (a_1, a_2, a_3, a_4), podobnie (2) (tylko w odwrotnej kolejności)

* przypadki (3) – (6) są symetryczne
* spójrzmy na pierwszy z nich:

$$a_1 < a_2 > a_3 > a_4$$

* odpowiadają mu 3 permutacje (bo nie znamy relacji a_1 z a_3 i a_4):

$$(a_1, a_4, a_3, a_2), (a_4, a_1, a_3, a_2), (a_4, a_3, a_1, a_2)$$

* Łącznie mamy zatem $2 + 4 \cdot 3 = 14$ permutacji, które spełniają założenia zadania
* stąd, każdy algorytm sortujący przez porównania w pesymistycznym przypadku będzie musiał wykonać $\geq \lceil \log 14 \rceil = 4$ porównania (aby zdecydować, którą permutację dostał na wejściu)

b) * zauważmy, że relacje (1) – (6) możemy podzielić na dwa rodzaje ze względu na zależności między trzema pierwszymi elementami – albo jest to ciąg monotoniczny, albo element środkowy jest ekstremum

(1) $a_1 < a_2 < a_3 < a_4$

(2) $a_1 > a_2 > a_3 > a_4$

(3) $a_1 < a_2 > a_3 > a_4$

(4) $a_1 < a_2 < a_3 > a_4$

(5) $a_1 > a_2 < a_3 < a_4$

(6) $a_1 > a_2 > a_3 < a_4$

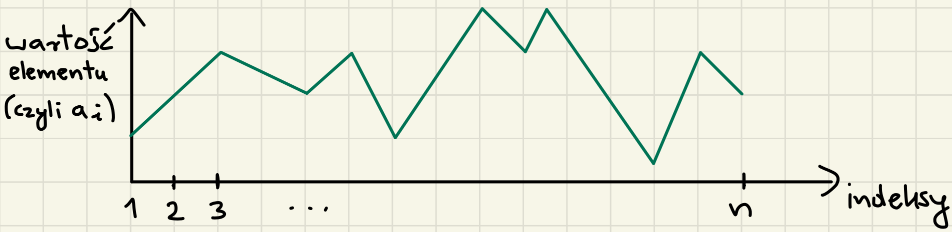
w przypadku wnioskujemy, że ciąg a_2, a_3, a_4 jest monotoniczny

* dlatego pierwsze dwa porównania używamy dla par (a_1, a_2) i (a_2, a_3)

* jeśli zachodzi , wstawiamy a_4 między a_1 i a_2, a dla – a_1 między a_3 i a_4

* w obu przypadkach dwa dodatkowe porównania wystarczą (stąd łącznie 4 porównania)

c) * tym razem analizujemy ciąg $(a_1, a_2, \ldots, a_n)$ n elementów, w którym są co najwyżej k ekstrema – to znaczy, że wartości elementów tego ciągu tworzą wykres podobny do poniższego



czyli ciąg składa się z co najwyżej k+1 uporządkowanych (monotonicznych) segmentów

* najpierw chcemy oszacować liczbę permutacji, które spełniają własność $\leq$ k ekstremów, na przykład dla $n = 8$, $k = 2$ permutacja

$$(a_1, a_5, a_2, a_4, a_6, a_3, a_7, a_8)$$

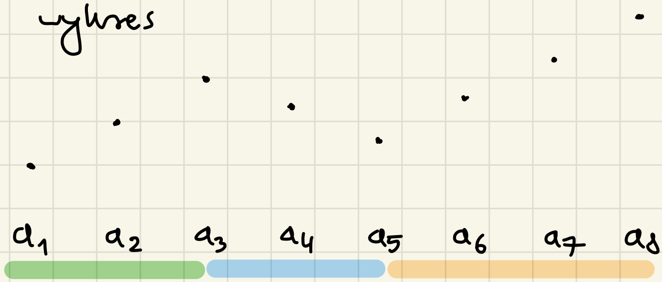
spełnia własność, bo możemy wyodrębnić trzy segmenty rosnące i dwa ekstrema

$$a_1 < a_5 < a_2 < a_4 < a_6 < a_3 < a_7 < a_8$$

* innymi słowy, dla relaji

$$a_1 < a_5 < a_2 < a_4 < a_6 < a_3 < a_7 < a_8$$

mamy wykres



* obserwaja: w permutaji muszą być ustalone relacje między ekstremami a ich sąsiadami oraz między elementami wewnątrz segmentów; kolejność między elementami z różnych segmentów nie ma wpływu na „wygląd" wykresu
* musimy teraz skorzystać z tej obserwaji, żeby pokazać, że istnieje „wystarczająco dużo" permutaji opisujących ciągi z $\leq k$ ekstremami
* rozpoczynamy od pytania, jakie oszacowanie wystarczy?

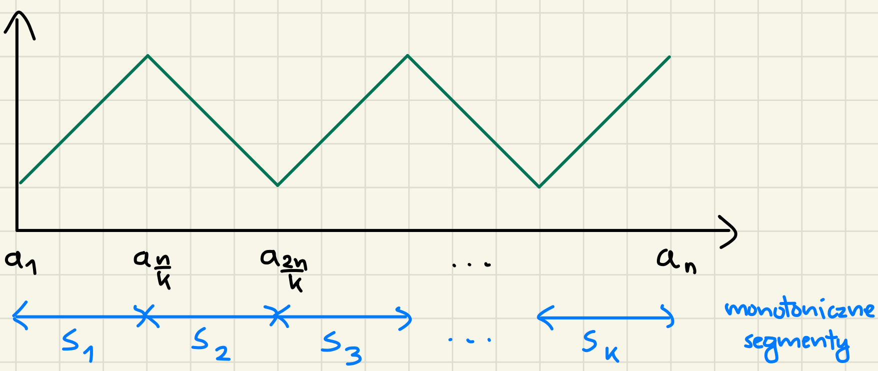
* intuicja: dla $k = n$ nie wiemy nic o ciągu, czyli złożoność wyniesie $O(n \log n)$, a dla $k = 1$ wystarczy $O(n)$ (szukamy ekstremum i jeśli istnieje, złączamy dwa uporządkowane ciągi)

* stąd „celujemy" w złożoność $O(n \log k)$

* wystarczy zatem pokazać, że istnieje $\geq (k!)^{\frac{n}{k}}$ permutacji, między którymi musimy wybierać, bo ze wzoru Stirlinga

$$\log (k!)^{\frac{n}{k}} = \frac{n}{k} \log(k!) \approx \frac{n}{k} \cdot k \log k = n \log k$$

* rozważmy ciąg z $k-1$ ekstremami występującymi w równych odstępach, który najpierw rośnie:



* niech segment S_i zawiera elementy o indeksach od $\frac{n}{k}(i-1)+1$ do $\frac{n}{k}\cdot i$ dla $i \in \{1, 2, \dots, k\}$

* zakładamy, że S_i rosnący dla i nieparzystych, a malejący dla parzystych, musimy więc zachować relacje:

$$a_1 < a_2 < a_3 < \dots < a_{\frac{n}{k}} \qquad (1)$$

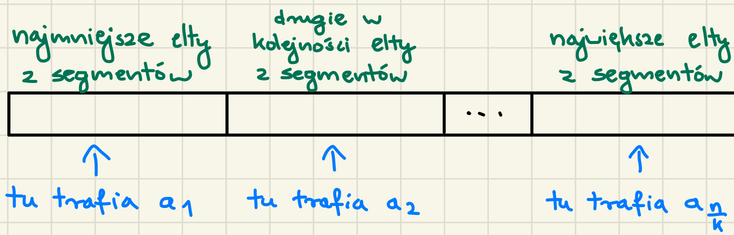
$$a_{\frac{n}{k}+1} > a_{\frac{n}{k}+2} > a_{\frac{n}{k}+3} > \dots > a_{\frac{2n}{k}} \qquad (2)$$

$$\dots$$

itd., ale co ważne, możemy nic nie mówić o relacji między $a_{i\frac{n}{k}}$ i $a_{i\frac{n}{k}+1}$ — łatwo sprawdzić, że z uporządkowania samych segmentów wynika, że jeden z tych elementów będzie ekstremum, choć nie wiemy który

* pozostaje zauważyć, że jeśli w permutacji na pierwszych k miejscach ustawimy w dowolnej kolejności najmniejsze elementy z segmentów, na drugich k pozycjach - drugie najmniejsze elementy itd., to zachowamy nierówności (1), (2), ..., (k)

* rzeczywiście, np. dla S_1 mamy:



czyli spełnione są nierówności (1) ← dla S_i tak samo

* pozostaje zauważyć, że w każdym k-elementowym bloku permutacji powyżej mamy $k!$ możliwości rozmieszczenia elementów, a bloków jest $\frac{n}{k}$

* stąd łącznie wygenerowaliśmy $(k!)^{\frac{n}{k}}$ permutacji, co było naszym celem

* na koniec podajemy algorytm o złożoności $O(n \log k)$ sortujący ciągi n-elementowe z co najwyżej k ekstremami:

  - przejdź przez ciąg, wyznaczając segmenty
  - stwórz kopiec i wrzuć do niego po najmniejszym elemencie z każdego segmentu (jest ich max $k+1$)
  - pobieraj min z kopca i dokładaj najmniejszy element, jeśli taki jeszcze jest, z segmentu, z którego było min

Uwagi końcowe:

* dokładny opis i analizę algorytmu sortującego pomijam – można to potraktować jako ćwiczenie
* warto porównać to zadanie z zadaniem o ciągach k-dobrych (omawianym na trzecich ćwiczeniach, numer zadania: 2.2)