

Algorytmy i Struktury Danych 2: Laboratorium 8

Przemysław Hołda

Koty

Zainspirowani opowieściami podróżnika Fryderyka o jego zmaganiach z numeracją dni na mapach i jego spotkaniu z tajemniczą żabą uciekająca przed bocianem postanowiliśmy udać się w podróż. Aczkolwiek, w domu mamy dużo kotów. Znając ich destruktywną naturę obawiamy się pozostawić je bez opieki. Zatem naszym celem będzie znalezienie im opiekunów. Jednakże, może się to okazać nie takie proste, gdyż każdy kot ma swoje preferencje i tylko wybrani opiekunowie mogą się nim zajmować. W dodatku, każdy opiekun może zająć się pewną maksymalną liczbą kotów. W pierwszej części zadania będziemy chcieli przekonać się czy możemy przekazać koty zadany opiekunom. W kolejnej części zadania opiekunowie będą od nas chcieli pieniądze za zajmowanie się naszymi zwierzętami. Dlatego interesować nas będzie minimalna kwota jaką musimy przeznaczyć za opiekę nad wszystkimi kotami.

Opis danych

W obu etapach dostajemy tablicę `cats`, która zawiera w sobie instancje struktury `Cat`. Struktura ta zawiera w sobie jedno pole o nazwie `AcceptablePeople`, które jest tablicą zawierającą identyfikatory osób, które kot akceptuje (innymi słowy – tylko te osoby mogą zajmować się danym kotem). Jako identyfikatory kotów przyjmujemy odpowiadające im indeksy w tablicy `cats`.

Następnie, dostajemy tablicę `people`, która składa się z potencjalnych opiekunów. Każdy opiekun jest instancją struktury `Person`, która zawiera dwa pola. Pierwsze to `MaxCats`, które oznacza maksymalną liczbę kotów jaką dana osoba może do siebie przyjąć. Drugie pole to `Salaries`. Zapewnia ono informację o należności jaką będziemy musieli zapłacić danej osobie za opiekę nad wybranym kotem. Jest to tablica liczb całkowitych, gdzie indeks to identyfikator kota, a otrzymana wartość to należna kwota.

W każdym zadaniu należy zwrócić informację czy istnieje możliwe przypisanie kotów do opiekunów. Dodatkowo, wymagane jest zwrócenie znalezione przypisania. Jest ono reprezentowane jako tablica, której indeksami są kolejne identyfikatory osób, natomiast wartościami są tablice składające się z identyfikatorów kotów (kolejność nie ma znaczenia). Ponadto, w drugim zadaniu trzeba zwrócić informację o minimalnym koszcie jaki przyjdzie nam zapłacić.

Pierwsze zadanie

W tym zadaniu chcemy znaleźć takie przypisanie, żeby każdy kot miał dokładnie jednego opiekuna. Należy pamiętać, że opiekun może zająć się wieloma kotami. Gdy będzie to możliwe należy w zwracanej krotce zwrócić `isPossible` ustawione na `true`. Następnie, przypisanie spełniające warunki zadania należy zwrócić w zwracanej krotce w miejscu `assignment`. W czasie weryfikowania pierwszego etapu przypisanie nie będzie sprawdzane (można zwrócić `null`).

Drugie zadanie

W tym zadaniu interesuje nas minimalny koszt przypisania kotów do opiekunów. Każdemu opiekunowi płacimy osobno za każdego kota, którego do siebie przyjął (zgodnie z odpowiadającym mu `Salaries`). Zatem celem minimalizacji jest suma poniesionych kosztów. Tak jak w pierwszym zadaniu kot musi być przypisany do dokładnie jednego opiekuna. Znalezione minimalny całkowity koszt należy zwrócić w krotce w miejscu `minCost`. Następnie, należy zwrócić przypisanie kotów do opiekunów zgodnie z opisem z pierwszego zadania. W razie znalezienia rozwiązania pamiętać o zwróceniu `isPossible` ustawionego na `true`. W czasie weryfikowania trzeciego etapu przypisanie nie będzie sprawdzane (może być ustawione jako `null`).

Przykład

Rozpatrzmy następujący przypadek. Tablica `cats` składa się z następujących kotów:

0. `Cat(acceptablePeople: [1, 2])`,
1. `Cat(acceptablePeople: [0])`,
2. `Cat(acceptablePeople: [0, 1])`,
3. `Cat(acceptablePeople: [1, 2])`,
4. `Cat(acceptablePeople: [2])`

Kolejne identyfikatory kotów to 0, 1, 2, 3 i 4. Przykładowo widzimy, że pierwszy kot chce, żeby opiekowała się nim osoba o identyfikatorze 1 lub 2. A drugi kot zaakceptuje tylko osobę 0. Następnie dostajemy tablicę `people`, której zawartość wygląda następująco:

0. `Person(MaxCats: 2, Salaries: [50, 100, 100, 100, 65])`,
1. `Person(MaxCats: 3, Salaries: [50, 80, 50, 50, 60])`,
2. `Person(MaxCats: 1, Salaries: [200, 120, 180, 200, 300])`

Kolejne identyfikatory osób to 0, 1 i 2. Pierwsza osoba może zająć się maksymalnie dwoma kotami. Za kota o identyfikatorze 0 dostanie 50, za kota o identyfikatorze 1 otrzyma 100, itd.

W pierwszym zadaniu moglibyśmy zwrócić następujące przypisanie: pierwsza osoba zajmie się kotami 1 i 2, druga osoba kotami 0 i 3, trzecia osoba kotem 4. Natomiast w drugim zadaniu: pierwsza osoba zajmie się kotem 1 (koszt 100), druga osoba kotami 0, 2 i 3 (koszt $50 + 50 + 50$), trzecia osoba kotem 4 (koszt 300) – minimalny koszt to 550.

Etapy

Etap 1 - pierwsze zadanie, zwrócić czy przypisanie jest możliwe (1 pkt)

Etap 2 - pierwsze zadanie, zwrócić możliwe przypisanie (0.5 pkt)

Etap 3 - drugie zadanie, zwrócić minimalny koszt przypisania (0.5 pkt)

Etap 4 - drugie zadanie, zwrócić przypisanie dla minimalnego kosztu (0.5 pkt)

Uwagi

1. Gdy istnieje wiele równoważnych przypisań zwrócić dowolne z nich.
2. Mamy gwarancję, że u każdej osoby **Salaries** będzie długości równej liczbie kotów, a każda wartość będzie dodatnia. Również wartość **MaxCats** będzie dodatnia.
3. W przypadku braku kotów przypisanie istnieje (dodatkowo, gdy tablica **people** jest pusta, to zwracane **assignment** będzie pustą tablicą).