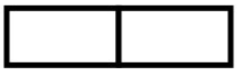
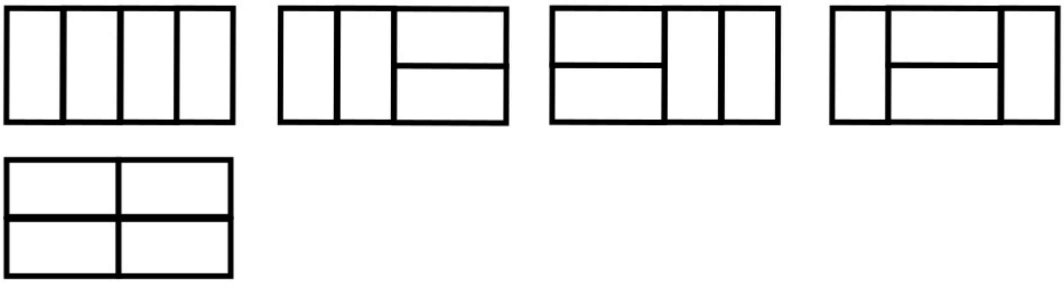
Podlaski Turniej w Programowaniu Zespołowym 2015
Rozwiązania niektórych zadań

Chodnik

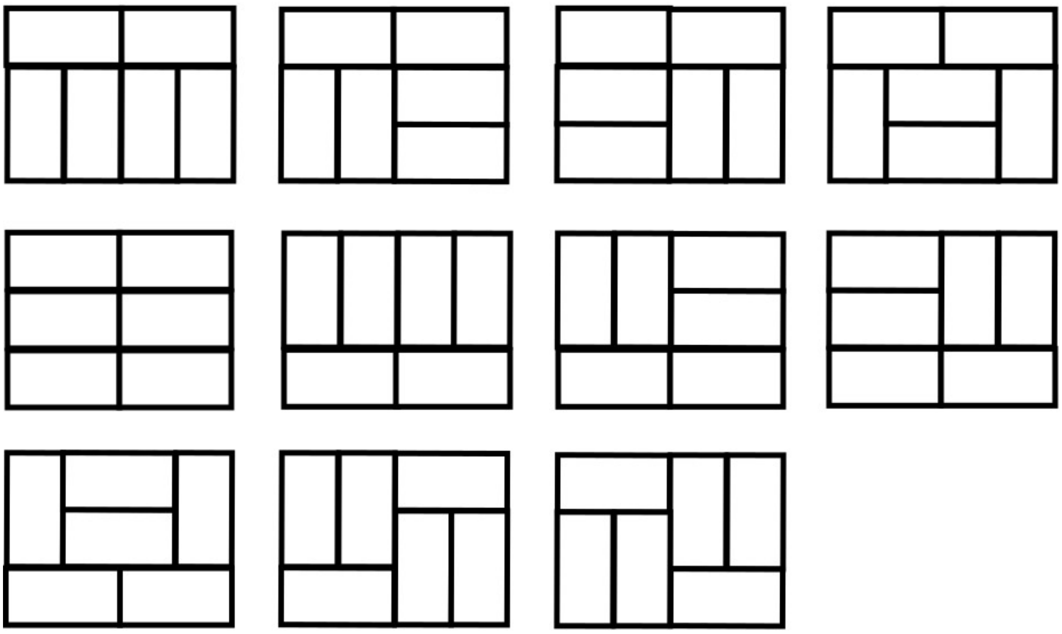
Zadanie polega na znalezieniu liczby sposobów (oznaczmy S_n) ułożenia chodnika o szerokości 4 i długości n mając do dyspozycji płyty chodnikowe rozmiaru 2 na 1. Na początku warto przyjrzeć się początkowym wartościom – istnieje tylko jeden sposób ułożenia chodnika o długości 1 (dwie poziome płyty obok siebie). W przypadku chodnika o długości 2 mamy już 5 sposobów jego ułożenia, natomiast w przypadku chodnika o długości 3 istnieje aż 11 sposobów. Oto te sposoby:



$S_1=1$

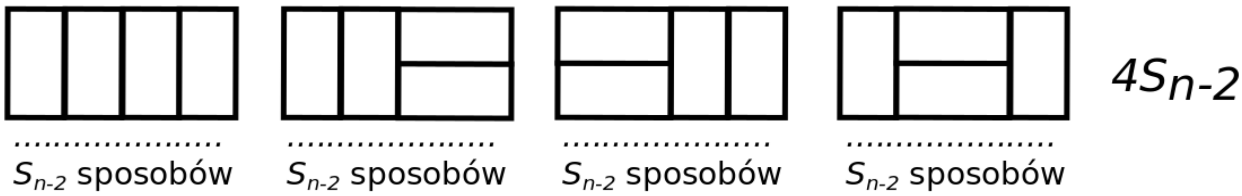


$S_2=5$

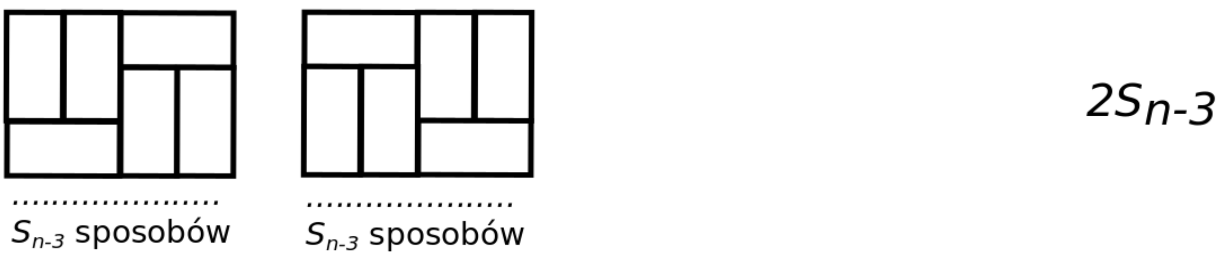


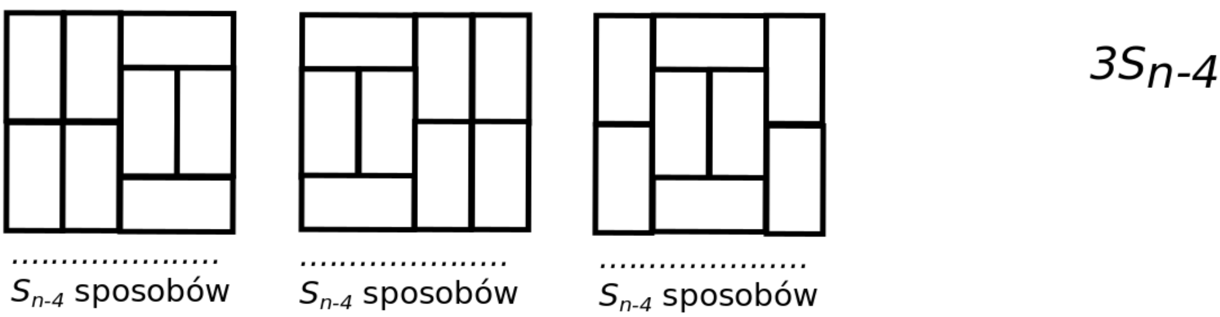
$S_3=11$

Rozwiązanie problemu sprowadza się do znalezienia wzoru rekurencyjnego na S_n. Aby znaleźć zależność S_n od poprzednich wyrazów ciągu należy rozważyć wszystkie możliwości zakończenia chodnika długości n. Przez zakończenie chodnika rozumiemy fragment bez wewnętrznych bruzd poziomych w całości go przecinających. Poniżej widzimy, że istnieje jedno zakończenie chodnika posiadające długość 1, a pozostałą część chodnika (o długości n-1) można ułożyć na S_{n-1} sposobów. Ponadto istnieją cztery zakończenia mające długość 2 – dla każdego z tych zakończeń pozostałą część chodnika (długości n-2) można ułożyć na S_{n-2} sposobów.

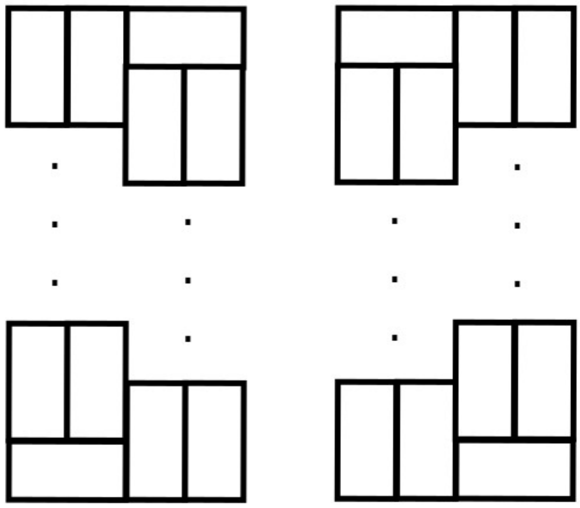


Rozważmy teraz zakończenia chodnika o długości 3 oraz o długości 4:

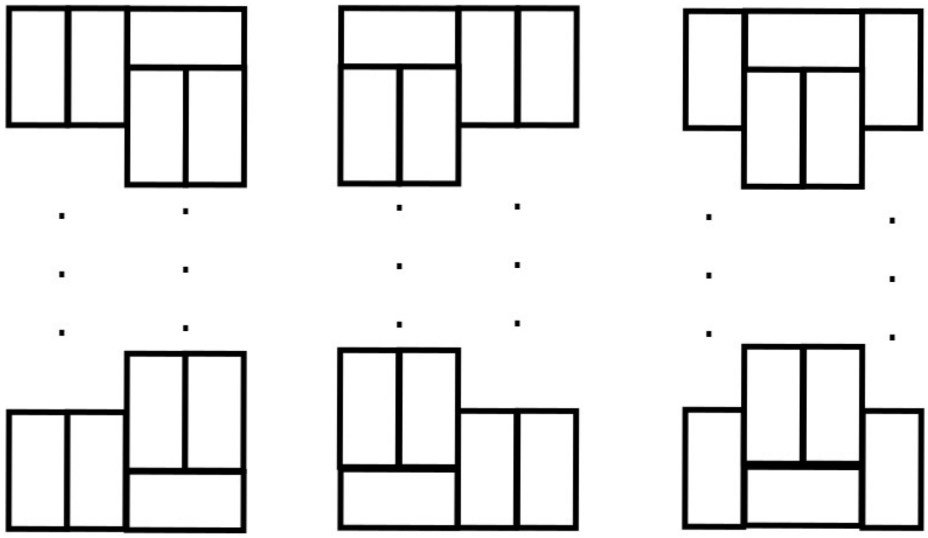




Widać, że istnieją tylko dwa zakończenia chodnika długości 3 oraz trzy zakończenia długości 4. Można zauważyć, że istnieją zawsze dwa różne zakończenia o nieparzystej długości (dłuższe niż 2) oraz trzy różne zakończenia o parzystej długości (dłuższe niż 2). Wynika to z faktu, że to jak zaczynamy budować kończący fragment od razu jednoznacznie wskazuje, jak ma on wyglądać. Oto schematy możliwych zakończeń nieparzystej i parzystej długości.



$2S_{n\text{-}k}$
(nieparzyste k>2)



$3S_{n\text{-}k}$
(parzyste k>2)

Dzięki powyższemu spostrzeżeniu uzyskujemy wzór rekurencyjny:

$$S_n=S_{n-1}+4S_{n-2}+2S_{n-3}+3S_{n-4}+2S_{n-5}+3S_{n-6}+\ldots$$

Zauważmy, że:

$$S_{n-2}=S_{n-3}+4S_{n-4}+2S_{n-5}+3S_{n-6}+\ldots$$

Odejmując od siebie oba równania stronami mamy:

$$S_n-S_{n-2}=S_{n-1}+4S_{n-2}+S_{n-3}-S_{n-4}$$

A więc ostatecznie: $\boldsymbol{S_n=S_{n-1}+5S_{n-2}+S_{n-3}-S_{n-4}}$

Aby wielokrotnie nie liczyć tych samych wyrazów dla kolejnych zapytań należy obliczyć raz wszystkie potrzebne wartości S_n i umieścić je w tablicy. Daje to akceptowalną złożoność O(n), gdzie n to maksymalna długość chodnika.

Kasa misiu, kasa

W zadaniu dany jest graf ważony z dwoma rodzajami wag powiązanymi z krawędziami: czasami przejazdu oraz opłatami za przejazd (liczby całkowite dodatnie). Problem polega na wyznaczeniu ścieżki o najkrótszym czasie przejazdu z punktu startowego *s* do punktu końcowego *e*, której łączna opłata nie przekroczy zadanego budżetu *B*. Zadanie znalezienia najkrótszych ścieżek z ograniczeniami jest swego rodzaju połączeniem klasycznego problemu najkrótszych ścieżek z innymi problemami optymalizacyjnymi. W rozwiązaniu zadania ważne jest spostrzeżenie, że najszybsze dotarcie z punktu startowego do punktu *k* , wydając przy tym dokładnie *p* pieniędzy zależy od najszybszego dotarcia do sąsiadów wierzchołka *k* przy wydaniu odpowiednio mniejszych sum pieniędzy. Niech t_{ij} oraz o_{ij} oznaczają odpowiednio czas trwania i opłatę dla podróży między miastami *i* oraz *j*. Niech $f(k,p)$ oznacza czas najszybszego dojazdu z wierzchołka startowego do wierzchołka *k* wydając przy tym dokładnie *p* pieniędzy. Niech $S=\{s_1, s_2, \ldots, s_z\}$ oznacza zbiór sąsiadów wierzchołka *k*, do których prowadzą krawędzie o opłacie nie większej niż *p*. Wtedy:

$$f(k,p)=min(f(s_1, p-o_{ks1})+t_{ks1}, f(s_2, p-o_{ks2})+t_{ks2}, \ldots, f(s_z, p-o_{ksz})+t_{ksz})$$

Wykorzystując powyższą własność optymalnej podstruktury należy zastosować programowanie dynamiczne przetwarzając rozwiązania problemów (najkrótsze trasy) w sposób wstępujący: od najmniejszych (nisko płatnych) do największych (o wysokiej łącznej opłacie). Najpierw mamy tylko jedno rozwiązanie dla wierzchołka startowego: $f(s,0)=0$. Mając optymalne rozwiązanie problemu najszybszego dotarcia do wierzchołka *k* płacąc *p* (czyli $f(k, p)$) wystarczy przeprowadzić relaksację dla wszystkich jego sąsiadów s_i do których można dotrzeć bez przekraczania maksymalnego, dostępnego budżetu:

$$f(s_i, p+o_{ksi})=min(f(s_i, p+o_{ksi}), f(k, p)+t_{ksi})$$

Dzięki takiej metodzie przetwarzania podproblemów łatwo można dokonać pewnej optymalizacji: relaksację sąsiadów wystarczy przeprowadzać tylko dla rozwiązań pareto-optymalnych (takie $f(k,p)$, że dla każdego $q<p$ mamy $f(k,q)>f(k,p)$). Innymi słowy nie musimy przetwarzać danego rozwiązania dla wierzchołka *k*, jeżeli już wcześniej udało się uzyskać nie gorsze rozwiązanie dla tego wierzchołka przy mniejszej łącznej opłacie. Pozwala to dodatkowo zredukować czas obliczeń. Finalne rozwiązanie dla wierzchołka końcowego *e* to $min(f(e,0), f(e,1), \ldots, f(e, B))$, a złożoność pesymistyczna to $O((|V|+|E|)B)$.

Bezpieczne liczby

W zadaniu należy wyznaczyć liczbę bezpiecznych liczb pierwszych należących do przedziału $<a, b>$. Liczba pierwsza *p* jest bezpieczna, gdy $\lfloor p/2 \rfloor$ też jest liczbą pierwszą. Ze względu na duży rozmiar przedziału (do 10^7) i duże rozważane liczby (do 10^{12}) standardowe rozwiązania (wyszukiwanie wyczerpujące czy pojedyncze sito Eratostenesa) nie mogły zmieścić się w limicie czasu . Idea rozwiązania bardziej optymalnego opiera się na dwóch, nieco zmodyfikowanych sitach Eratostenesa z przedziałów $<a, b>$ oraz $<a/2, b/2>$. Sprawdzenie odpowiednich pozycji w obu sitach pozwala szybko stwierdzić, czy dana liczba jest bezpieczną liczbą pierwszą. Do „przesiewania” w tych sitach wystarczy użyć liczb pierwszych z przedziału $<2, 10^6>$ (górny zakres tego przedziału to pierwiastek z 10^{12}) - te można wyznaczyć szybko za pomocą standardowego sita. Daje to rozwiązanie o akceptowalnej złożoności $O((b-a)*\ln \ln(b))$.

Opracował Krzysztof Ostrowski