Розбір

Потоки в мережі і суміжні задачі

A. юра і тренування 1

Коротка умова

- є неорієнтований граф з n <= 500 вершин і m <= 1000 ребер
- ребра мають пропускну здатність a <= 10^6
- потрібно знайти максимальний потік у графі

А. розв’язок

Знаходимо максимальний потік

(алгоритм Едмондса-Карпа,

або будь-який інший).

B. юра і тренування 2

Коротка умова

- є неорієнтований граф з n <= 1000 вершин і m <= 3000 ребер
- ребра мають пропускну здатність a <= 10^5
- потрібно знайти максимальний потік у графі

B. розв’язок

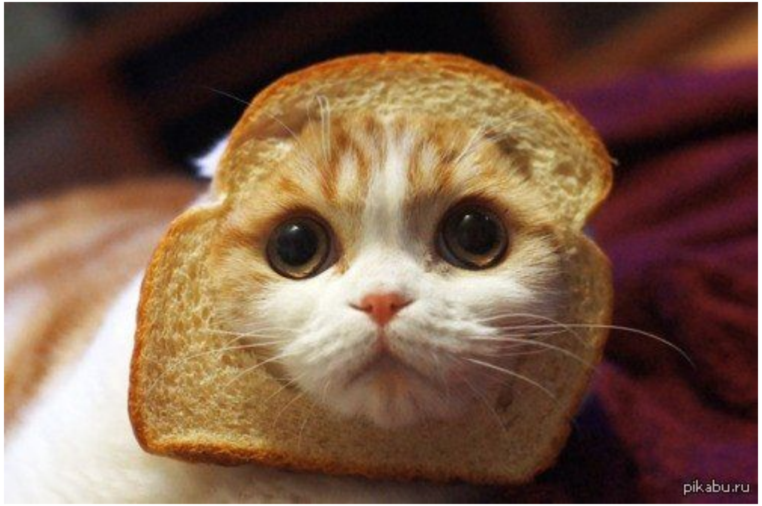
Знаходим максимальний потік

за допомогою кращих способів

(алгоритм Дініца,

або проштовхування

предпотоку).



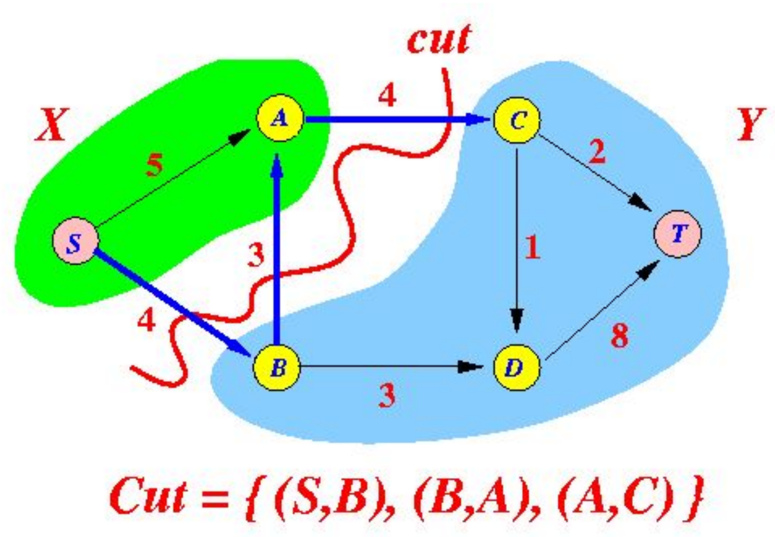
С. юра і тренування 3

Коротка умова

- є планарний граф з n <= 100000 вершин
- виток - найлівіша вершина
- стік - найправіша
- ребра мають пропускну здатність a <= 10^3
- потрібно знайти максимальний потік у графі

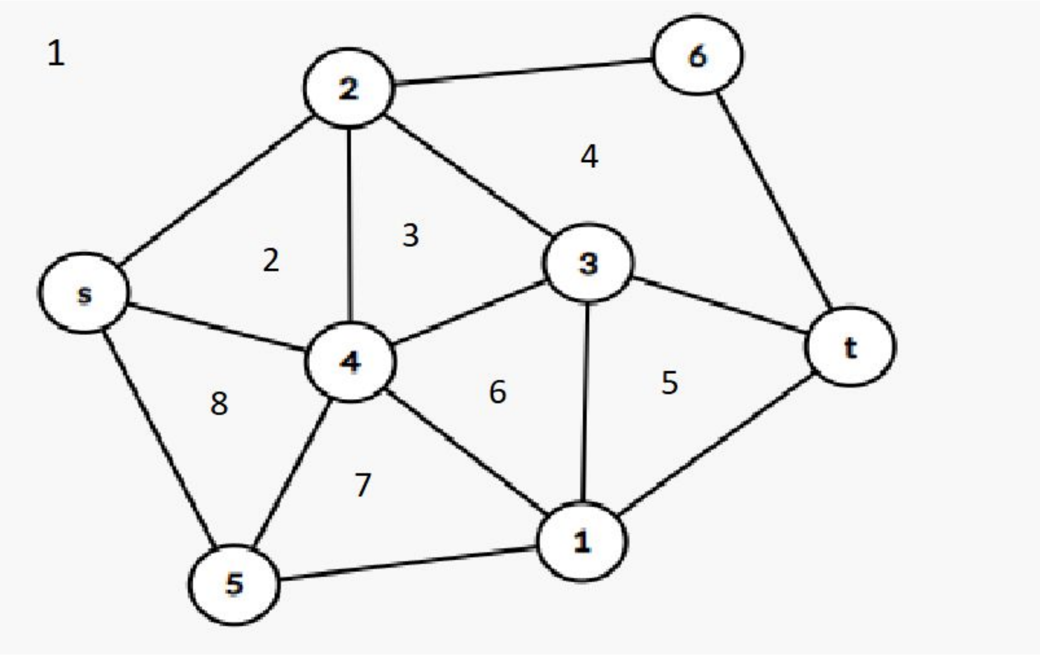
С. розв’язок

- замість максимального потоку можна шукати мінімальний розріз



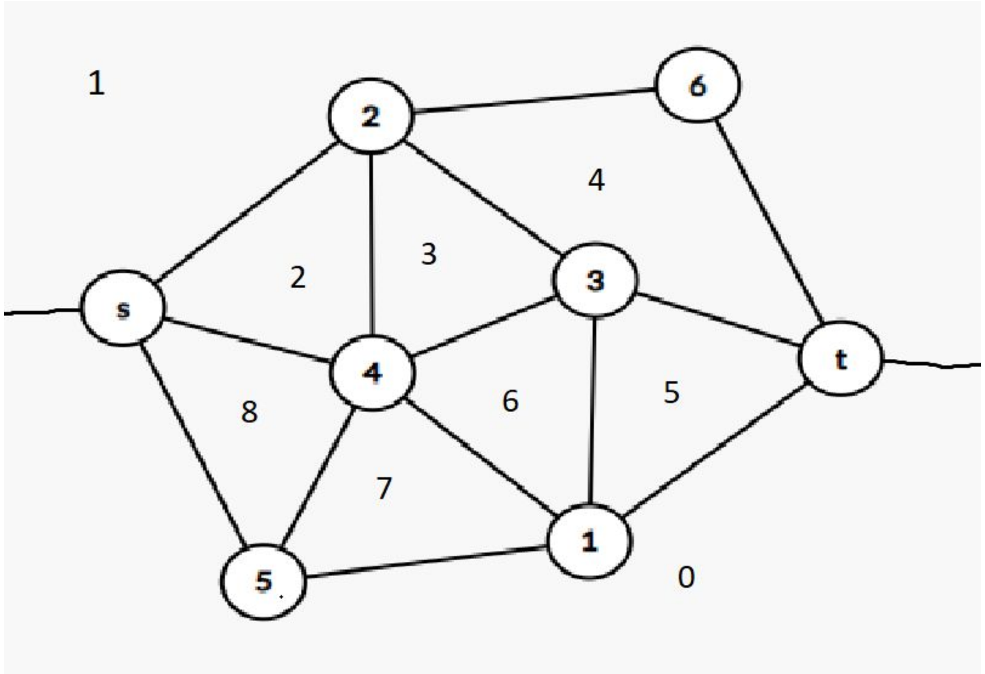
C. розв’язок

- знайдемо грані планарного графа



C. розв’язок

- розіб’ємо зовнішню грань на дві частини

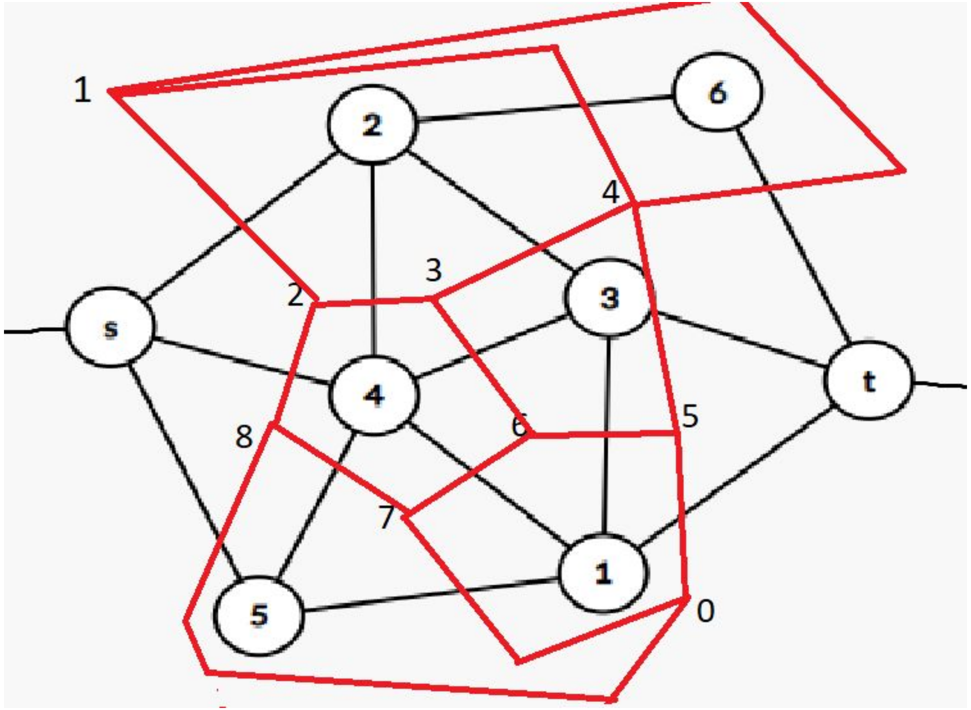


C. розв’язок

- прибираючи ребро графа ми з’єднуємо дві грані
- якщо верхня і нижня грань графа не з’єднані, то видалених ребер не достатньо, щоб розрізати граф на дві частини (існує шлях від s до t)
- потрібно видалити ребра, сума пропускних здатностей яких мінімальна

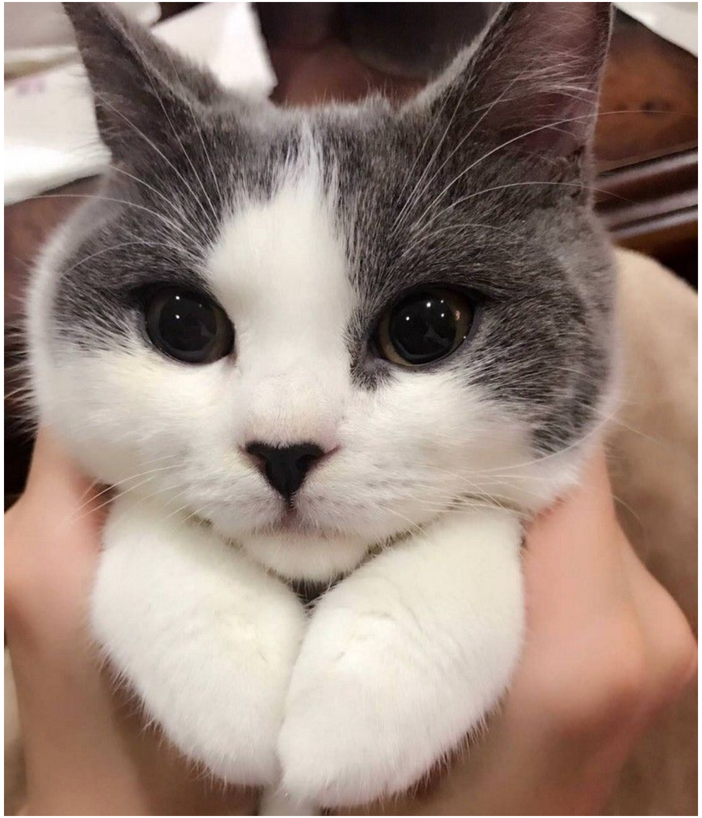
С. розв’язок

- побудуємо новий граф в якому ребра будуть з’єднувати грані.
- якщо в початковому при видаленні ребра грані з’єднувались, то в новому між цими гранями буде ребро ціна якого рівна пропускній здатності початкового ребра.



С. розв’язок

- відповідь - довжина найкоротшого шляху між верхньою та нижньою гранями (шукаємо за допомогою Дейкстри)



D. обід

короткa умова

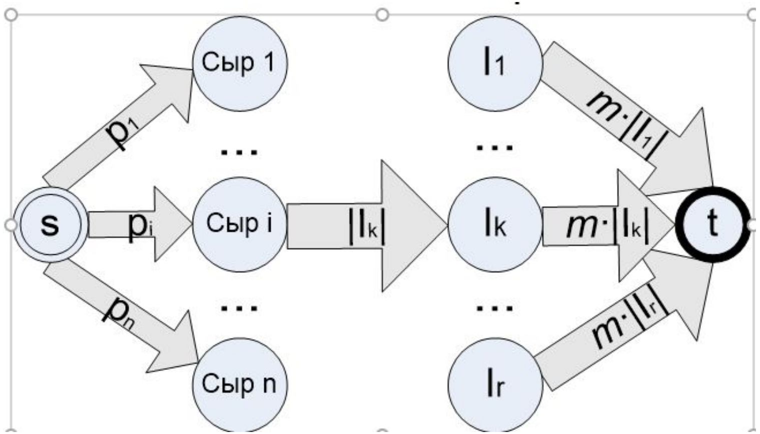
- є страви, які з’являються в певний час і в певний час перестають бути свіжими
- нам відомо вагу кожної страви
- є працівники і кожен працівник їсть їжу з певною швидкістю
- два працівники не можуть їсти одну страву одночасно
- знайти мінімальний час, при якому їжа вже не свіжа, але її продовжують їсти

D. розв’язок

- бінарний пошук по відповіді
- для кожної страви є інтервал часу впродовж якого її можна їсти
- потрібно перевірити, чи можливо з’їсти все

D. розв’язок

- візьмемо всі початки та кінці інтервалів та відсортуємо
- візьмемо два сусідні моменти часу в масиві
- впродовж цього інтервалу кожну страву або можна їсти, або не можна
- якби всі працівники їли з швидкістю 1, то задачі можна було б розв’язати за допомогою пошуку максимального потоку в наступному графі

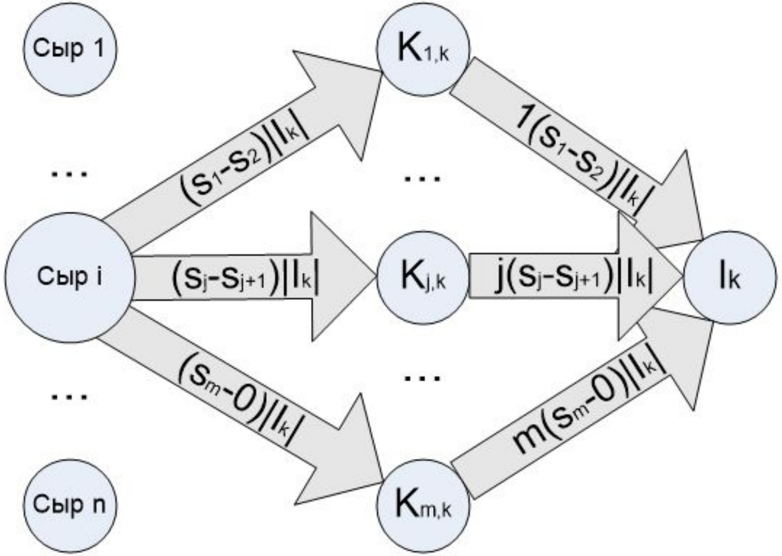


D. розв’язок

- відсортуємо всіх працівників за спаданням швидкості поїдання s[i]
- розібємо передостаннього працівника на двох (зі швидкістю поїдання s[m] та s[m] - s[m-1])
- аналогічно розіб’ємо інших працівників (1 буде розбитий на m)
- ми отримали m груп працівників із швидкостями поїдання (s[m], s[m] - s[m-1], s[m-1], s[m-2], … s[2] - s[1])
- в першій групі m працівників, в другій m-1, …

D. розв’язок

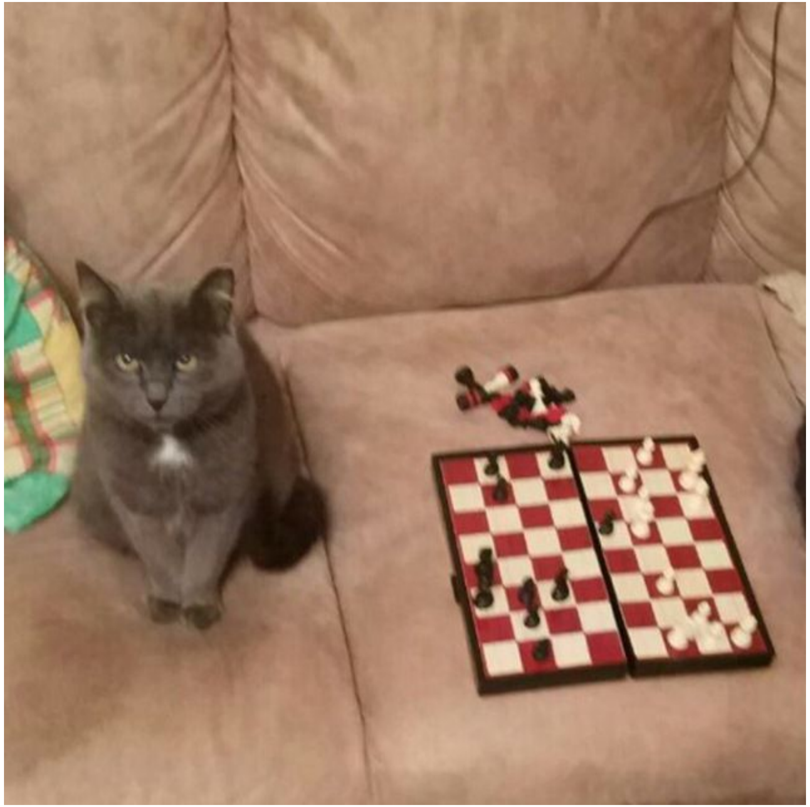
- кожна страва в будь-який момент часу поїдається не більше, ніж одним працівником з кожної групи
- кожна група поїдає не більше, ніж кількість працівників в ній, помножена на швидкість групи, грам їжі в годину
- будуємо наступну мережу і шукаємо в ній максимальний потік
- K[j][k] - одні й ті самі для різних страв і для одного інтервалу



E. шахи

коротка умова

- є шахова дошка n*m
- деякі клітинки заборонені
- потрібно розмістити на ній максимальну кількість конів, щоб вони не били один одного



Е. розв’язок

- кінь із білої клітинки завжди переходить на чорну, а із чорної на білу
- побудуємо дводольний граф
- потрібно знайти максимальну кількість вершин серед яких через жодні дві не проходить ребро (максимальну незалежну множину)
- ця величина рівна кількості вершин графа - розмір мінімального покриття
- розмір мінімального покриття рівний максимальному паросполученню
- максимальне паросполучення можна шукати за допомогою потоків в мережі або алгоритму Куна

F. день народження Стаса

Коротка умова

- дано неорієнтований граф
- потрібно видалити ребра, щоб степінь кожної вершини була непарна

F. розв’язок

- для кожної компоненти будемо розв’язувати окремо
- якщо кількість вершин в компоненті непарна - це зробити неможливо
- навчимось розв’язувати задачу для дерева
- підвісимо дерево за будь-яку вершину
- для вершин, розмір піддерева яких непарний проведемо ребро до предка, а для всіх інших - ні
- вершини з парним розміром піддерева мають непарну кількість синів з непарним розміром піддерева (їхня степінь буде непарною)
- вершини з непарним розміром піддерева мають парну кількість синів з непарним розміром піддерева (їхня степінь теж буде непарною (парна кількість ребер від синів + одне до батька))

G. розміщення

Коротка умова

- дана перестановка чисел 1, 1, 2, 2, …, n, n
- потрібно знайти таку перестановку, щоб мінімальна відстань між двома однаковими елементами була максимальна і сума відстаней між старими і новими позиціями - мінімальна

G. розв’язок

- помітимо, що нам підходять тільки ті перестановки, в яких відстань між двома однаковими елементами рівна n, тобто положення i в першій половині масиву однозначно задає положення i в другій.
- будуємо повний двудольний граф
- ціна ребра із u в v рівна відстані abs(pos_1[u]-v) + abs(pos_2[u]-(v+n)) де pos_1[u] - перша позиція u в початковому масиві, а pos_2[u] - друга
- знайдемо мінімальне зважене повне паросполучення

Н. гра

Коротка умова

- дане поле 3*4
- між квадратами поля розміщені монети (17 переходів між 12 вершинами)
- при переході через сторону забирається більша половина монет
- якщо на стороні нема монет - переходити не можна
- потрібно зібрати якомога більше монет

H. розв'язок

- побудуємо граф в якому квадрати - вершини, сторони - ребра
- розіб'ємо кожне ребро графа на декілька нових, і-те з яких буде мати ціну - кількість монет які ми отримаємо при і-тому проходженні через сторону.
- Нам потрібно видалити з графа деяку множину ребер, так щоб залишився ейлеровий шлях і пройтись по ньому.

H. розв’язок

- не має сенсу видаляти більше одного ребра, що відповідає певній стороні
- видаляти завжди будемо тільки найдешевше ребро для сторони
- переберемо всі випадки видалити множину ребер
- для кожного випадку знайдемо компоненти зв’язності в яких буде ейлеровий шлях і виберемо ту, яка має найбільшу суму ребер