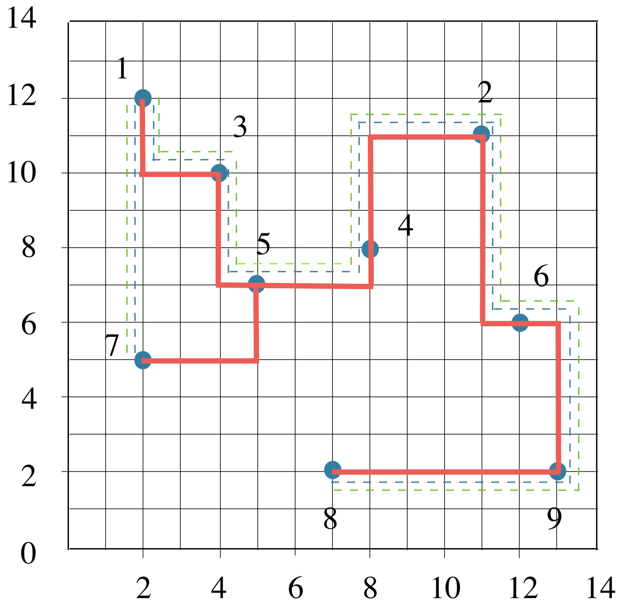
Приклад на побудову НЗС

На площині в декартовій системі координат задано місцезнаходження дев’яти виводів (як визначено в задачі). Відстань між будь-якими двома виводами m_i та m_j (як в алгоритмі) будемо визначати як $d_{ij} = \left| x_i - x_j \right| + \left| y_i - y_j \right|$.

Потрібно для заданої множини виводів $\{M\}$ визначити мінімальне зв’язуюче дерево та його довжину при таких обмеженнях

$\forall m_i \in M \left[\rho(m_i) \right] \leq 3 \quad i = \overline{1,9}$

Підкреслюю 1 … 9 - це множина виводів якихось КЕ (це **не** КЕ). Ці контакти потрібно з’єднати.



$d_{13}, d_{35}, d_{54}, d_{57}, d_{42}, d_{26}, d_{69}, d_{98}$

$d_{13}, d_{35}, d_{54}, d_{42}, d_{26}, d_{69}, d_{98}, d_{17}$

Подувати ініші варіанти та пакази чи є **НЗС** < 40. Можна побудувати багато різних по формі НЗС, але всі вони будуть мати $L_{НЗС} = 40$.

Визначемо відстань між виводами та складемо матрицю відстаней.

$d_{12} = (x_2 - x_1) + (y_1 - y_2) = (11-2) + (12-11) = 10$

$d_{13} = (x_3 - x_1) + (y_3 - y_1) = (4-2) + (12-10) = 4$

….

$$
D = \begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & 10 & \boxed{4^{1}} & 10 & 8 & 16 & 7 & 15 & 21 \\
2 & 10 & 0 & 8 & 6 & 10 & \boxed{6^{6}} & 15 & 13 & 11 \\
3 & 4 & 8 & 0 & 6 & \boxed{4^{2}} & 12 & 7 & 11 & 17 \\
4 & 10 & \boxed{6^{5}} & 6 & 0 & 4 & 6 & 9 & 7 & 11 \\
5 & 8 & 10 & 4 & \boxed{4^{3}} & 0 & 8 & \boxed{5^{4}} & 7 & 13 \\
6 & 16 & 6 & 12 & 6 & 8 & 0 & 11 & 9 & \boxed{5^{7}} \\
7 & 7 & 15 & 7 & 9 & 5 & 11 & 0 & 8 & 14 \\
8 & 15 & 13 & 11 & 7 & 7 & 9 & 8 & 0 & 6 \\
9 & 21 & 11 & 17 & 11 & 13 & 5 & 14 & \boxed{6^{8}} & 0
\end{array}
$$

- Аналізуємо перший рядок $|D|$ та вибираємо елемент $d_{13} = \min$ в цьому першому рядку. Якщо в рядку декілька елементів з рівними мінімальними значеннями, то вибираємо елемент з меншим значенням індексу j - тобто перший (a_{ij}) $j = \min$. Запам'ятовуємо елемент d_{13}; $\rho[1] = \rho[3] = 1$
- Викреслюємо елементи першого та третього стовпчиків матриці $|D|$
- Аналізуємо перший та третій рядки. Вибираємо $d_{35} = \min;$ $\rho[3] = 2, \rho[5] = 1$
- Викреслюємо елементи **п'ятого** стовпчика. Аналізуємо **перший**, **третій** та **п'ятий** рядки.
- Вибираємо $d_{54} = \min;$ $\rho[5] = 2, \rho[4] = 1$
- Викреслюємо четвертий стовпчик. Аналізуємо 1, 3, 5 та 4 рядки.
- Вибираємо $d_{57} = \min;$ $\rho[5] = 3, \rho[7] = 1$. Виключаємо з розгляду елементи сьомого стовпчика та п'ятого рядка, **тому що** $\rho[5] = 3$.

Продовжуємо побудову НВЗ, як розглядалося вище та вибираємо елементи.
Аналізуємо 1, 3, 4 та 7 рядки. $d_{42} = d_{46} = 6 = \min$. Вибираємо $d_{42} = 6 = \min$, як елемент з меншим **j**.

Аналізуємо 1, 2, 3, 4, 7, 2. $d_{26} = d_{46} = 6 = \min$. Вибираємо $d_{26} = 6 = \min$, як елемент з меншим **i**.

Аналізуємо 1, 2, 3, 4, 7, 2, 6. $d_{69} = 5 = \min$.
Аналізуємо 1, 2, 3, 4, 7, 2, 6, 9. $d_{98} = 6 = \min$.

Мінімальне зв'язуюче дерево представлено на малюнку (вище).
Загальна довжина його ребер дорівнює 40:
$$L_{\text{НВЗ}} = d_{13} + d_{35} + d_{54} + d_{57} + d_{42} + d_{26} + d_{69} + d_{98} = 4 + 4 + 4 + 5 + 6 + 6 + 5 + 6 = 40$$

Як альтернатива. Проведемо аналіз матриці, починаючи з 9 виводу.
Почнемо будувати з 9 рядка (9 вивода) - $d_{98} = 6 = \min$.
Викреслюємо 9 та 8 стовпчики; вибираємо $d_{96} = 5 = \min$.
Викреслюємо 6 стовпчик. Аналізуємо 9, 8, 6 рядки; вибираємо $d_{62} = 6 = \min$.
Викреслюємо 2 стовпчик. Аналізуємо 9-8-6-2 рядки; вибираємо $d_{24} = 6 = \min$ (по **i** **min**, $d_{64} = 6 = \min$ теж, але **i** велике)
Аналізуємо рядки 9-8-6-2-4; вибираємо $d_{45} = 4 = \min$.

Аналізуємо рядки 9-8-6-2-4-5; вибираємо $d_{53} = 4 = \min$.
Аналізуємо рядки 9-8-6-2-4-5-3; вибираємо $d_{57} = 5 = \min$
Аналізуємо рядки 9-8-6-2-4-5-3-7; вибираємо $d_{31} = 4 = \min$
Аналізуємо рядки 9-8-6-2-4-5-3-7-1

Таким чином побудова мінімального зв'язуючого дерева для ланцюга, що містить n виводів може бути описаний наступним чином:

- для довільного виводу ланцюга знайти найближчий (аналізуючи відповідні рядки) та привести з'єднання двох виводів.
- на кожних послідуючих кроках $i = 2, 3, \ldots, n-1$ з множини непід'єднаних виводів вибрати той, що знаходиться бліжче інших до групи вже зв'язаних виводів та під'єнати його до цієї групи по найкоротшому шляху (тобто аналізуємо рядкивиідповідних з'єднаних ребрами виводи

Побудоване таким чином дерево буде мати мімнімальну загальну довжину з'єднань виводів.

Тепер в термінах теорії графів початкова задача зводиться до визначення в повному графі **G** дерева, що включає всі вершини (виводи) та що має мімнімальну сумарну вагу ребер.

Таке дерево в різній літературі називають **мінімальним покриваючим деревом** - **НВЗ** - **деревом Пріма**.

Виникає питання - як змінює значення (величину) ρ на НВЗ - форму, довжину.

Давайте це розглянемо
Тобто $\rho \le 2$
Тоді все повторюється
d_{13}, d_{35}, d_{54} - далі 5 рядок викреслюємо , потім $d_{42}, d_{26}, d_{69}, d_{98}, d_{17}$
В першому рядку залишився один елемент що є невикресленним d_{17}
$L_{\text{НВЗ}} = d_{13} + d_{35} + d_{54} + d_{42} + d_{26} + d_{69} + d_{98} + d_{17} = 42$

Таким чином ρ впливає на форму та на довжину. Можна стрверджувати, що збільшення ρ дозволяє зменшити $L_{\text{НВЗ}}$ та зробити провідник більш складним по формі.

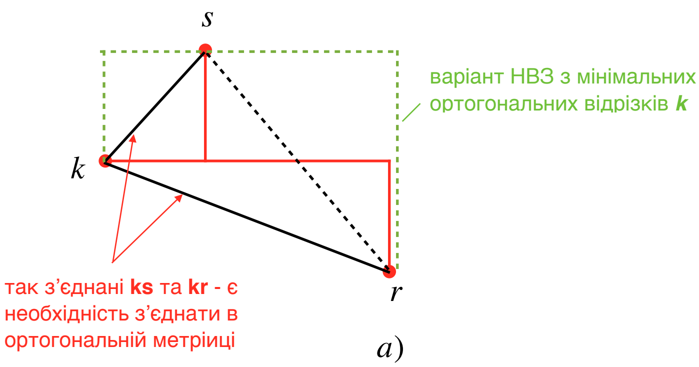
Провідник  для $\rho \le 2$

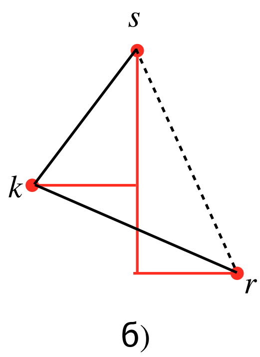
З рисунка слідує, що $\rho = 1$, тобто $\rho_1 = \rho_7 = 1$, а всі інші виводи мають $\rho = 2$

Отримавши НВЗ по Пріму *F* виконують перехід до квазіоптимальної сітки F^*
Задача побудови F^* зводиться до визначення сукупності мінімальних ортогональних підсистем, що зв'язують контакти у всіх створених трикутних підграфах та під'єднанню окремих контактів до відповідних контактів ортогональними з'єднаннями мінімальної довжини.

Розглянемо довільний трикутний підграф з вершинами ***s***, ***k***, ***r***.

Існують дві мінімальні ортогональні сітки зв'язуючі ці вершини - показані на рис.





Не важно доказати, що сумарна довжина сітки у них співпадає.

а) Для побудови даної мінімальної сітки визначається така вершина ***k***, проходяча через яку горизонтальна лінія, пересікає протилежну сторону трикутного підграфа.

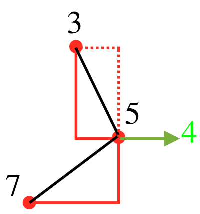
Опускаючи перпендикуляр з залишившихся вершин (***s*** та ***r***) на цю горизонталь, отримаємо мінімальну сітку, що зв'язує вершини трикутного підграфа шукаємою вершиною є вершина ***k***, якщо $x_k \le x_s \le x_r$ або $y_s \ge y_k \ge y_r$

$$L_a = x_{ks} + x_{sr} + y_{ks} + y_{kr}$$
$$L_b = x_{ks} + x_{sr} + y_{ks} + y_{kr}$$

тобто $L_a = L_b$

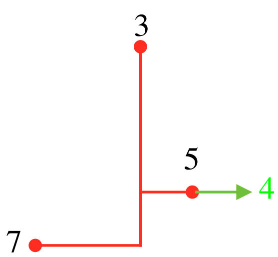
Для нашого прикладу

1) з'єднаємо 5-3, 5-7.



Побудуємо ортогональні ребра зв'язуючі виводи. З **5** проводимо горизонталь, яка пересікає протилежну сторону трикутника. Опустимо перпендикуляр з **3** та **7**. Отримаємо зв'язуючу сітку.

2) з 3 опускаємо вертикаль. З **5** та **7** будуємо перпендикуляри та отримаємо



Змінилася тільки форма, але $L_1 = L_2$

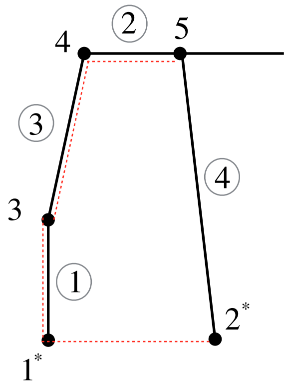
В деяких випадках, окрім обмеження на степені вершин зв'язуючого дерева, задається початкова та кінцева точка ланцюга.
Наприклад, це має місце при розробці монтажних схем для в/ч ланцюгів, коли необхідно зв'язати в визначеній послідовності джерело сигналу та декілька навантажень.

Тоді задача зводиться до побудови найкоротшого шляху між двома заданими виводами, що проходить через всі виводиланцюга, що залишилися.

Дана задача споріднена до задачі про маршрут комівояжера, але відрізняється тим, що шлях обходу повинен бути розімкнутим та з'єднувати дві задані точки, тобто виникає задача побудови **найкоротшого ланцюгу між заданими початкової та кінцевої вершинами**.

Розглянемо алгоритм, який наближенно вирішує цю задачу. Основу алгоритму складає (n - 1) кроковий процес вибору найкоротших ребер в повному графі **G**, перевірки кожного ребра на виконання обмеженної задачі та складання з вибраних ребер шляху, що з'єднує задані точки.

Нехай задано розташування точок.



5 точок

1^* та 2^* - початкові точки

1 2 3 4 послідовність з'єднання точок

- 1^* та 2^* - відповідно початкова та кінцева точки
- Складемо впорядковану по зростнню довжин послідовність ребер повного графу G :

$\left(1^*-3\right)$, $\left(1^*-2^*\right)$, $\left(2^*-3\right)$, $\left(4-5\right)$, $\left(3-4\right)$, $\left(3-5\right)$, $\left(1^*-4\right)$, $\left(2^*-5\right)$, $\left(2^*-4\right)$, $\left(1^*-5\right)$.

Це всеодно, що побудували повний граф та склали список ребер зростаючої довжини.

Чергове ребро $i=1,2,...,n-1$ вибирається по порядку з цюєї послідовності при виконанні наступних умов:

1. Ребро не з'єднує задані кінцеву та початкові точки (1^* та 2^*)
2. При включені ребра до шляху сетпінь вершин, що з'єднуються цим ребром, не перевищує допустимоі ($\rho=1$ для початкової та кінцевої точок та $\rho=2$ для інших точок).
3. Ребро не створює циклу з ребрами, що вже включені до шляху.
4. При включені до шляху будь-якого ребра, крім $n-1$ - го початкова та кінцеві точки залишаються не зв'язщаними

Умови 1-3 витікають з обмежень задачі. Умова 4 перешкоджає створенню тупікових ситуацій, тобто такого положення, при якому подальше формування шляху стає неможливим - всі прєднанні точки, крім початкової та кінцевої, мають $\rho=2$.

Формування шляху виконюється наступним чином:

1. Вибираємо ребро 1^*-3, тому що воно задовольняє всі вимоги - саме коротке, не з'єднує 1^* та 2^*, не створює цикл
2. Ребро 1^*-2^* відкидається, тому що не задовольняє умову 1, я ребро 2^*-3 не приєднуємо, тому що не задовольняє умову 4, тобто поки не побудоване дерево 1^* та 2^* залишаються незв'язаними. Вибираємо ребро $4-5$, тобто побудова НВЗ розпаралелилася.

3. Виббираємо ребро $3-4$
4. Ребра $3-5$ та $1^{*}-4$ відкидаються, тумощо створюються цикли. Вибирається $2^{*}-5$

Результуючий шлях $1^{*}-3-4-5-2^{*}$
Можна відмітити, що якщо зняти обмеження про крайні точки, то даний алгоритм приведе до більш короткого шляху $2-1-3-4-5$, тобто це чистий алгоритм Пріма.
Таким чином змінилася форма НВЗ та довжина. Довжина зменшилася.