A*:
informed search

С. Высоцкий

НАШ КУРС

Большие идеи в компьютерных науках и AI

Связь науки и практики в AI и CS

Практическая проблема

Теоретический фреймворк

Теоретическое решение

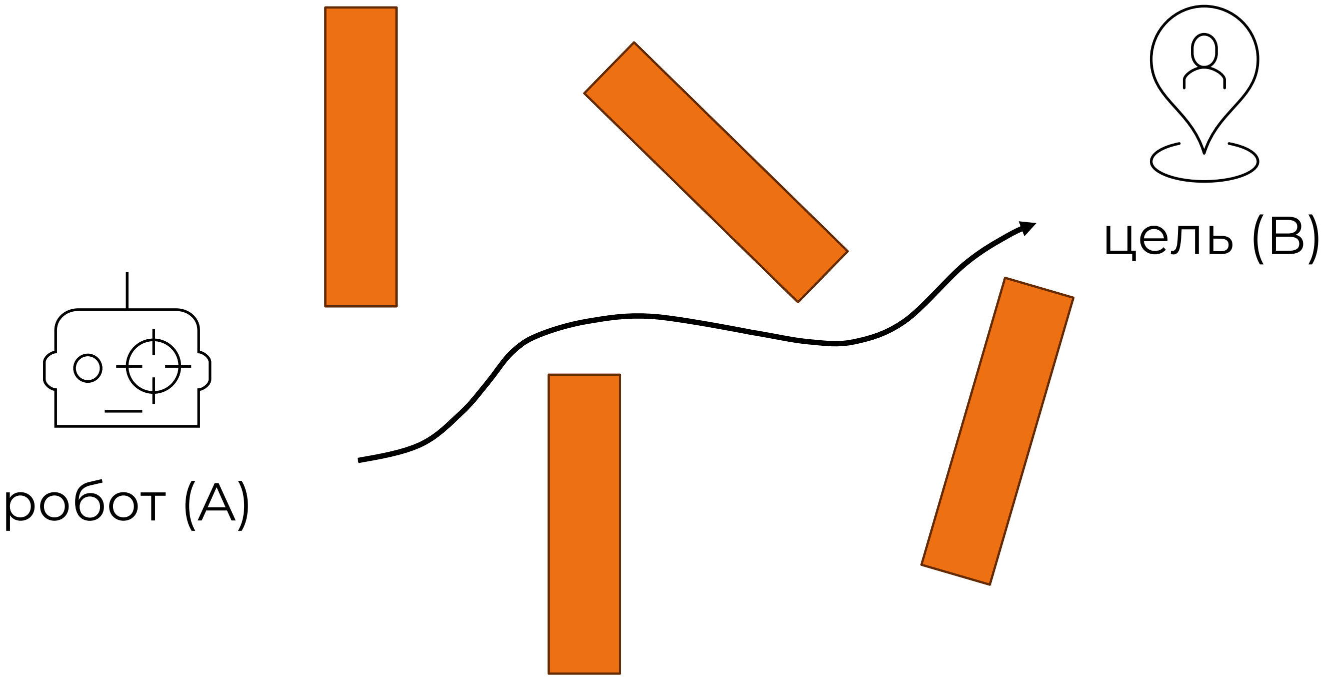
Адаптация

SHAKEY THE ROBOT

«The first general-purpose mobile robot able to reason about its own actions»

ПОИСК ОПТИМАЛЬНОГО ПУТИ

– препятствия



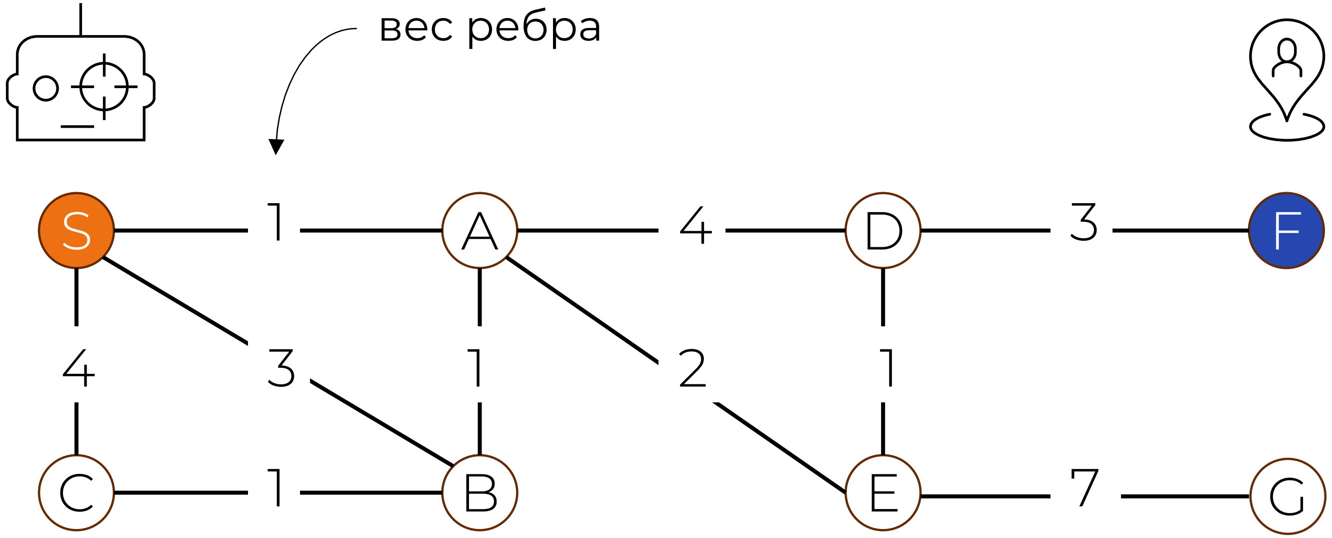
Задача

Перейти из **состояния** А в **состояние** В с **наименьшими затратами**

ДИСКРЕТНЫЙ СЛУЧАЙ

Задача на графе

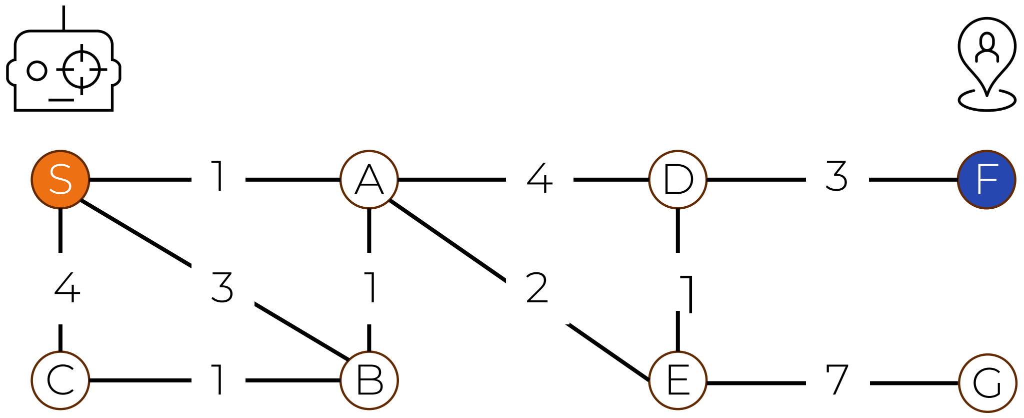
Найти путь между двумя вершинами с минимальным суммарным весом[1]

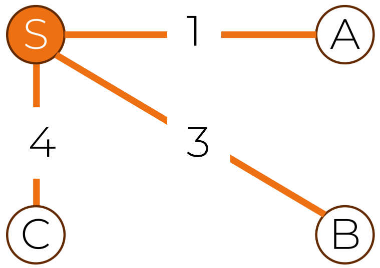


[1]Ребра имеют неотрицательный вес

Uninformed search

ЧТО МЫ ЗНАЕМ В САМОМ НАЧАЛЕ?





Из S можно добраться в A, B и C

$S \xrightarrow{1} A$

$S \xrightarrow{3} B$

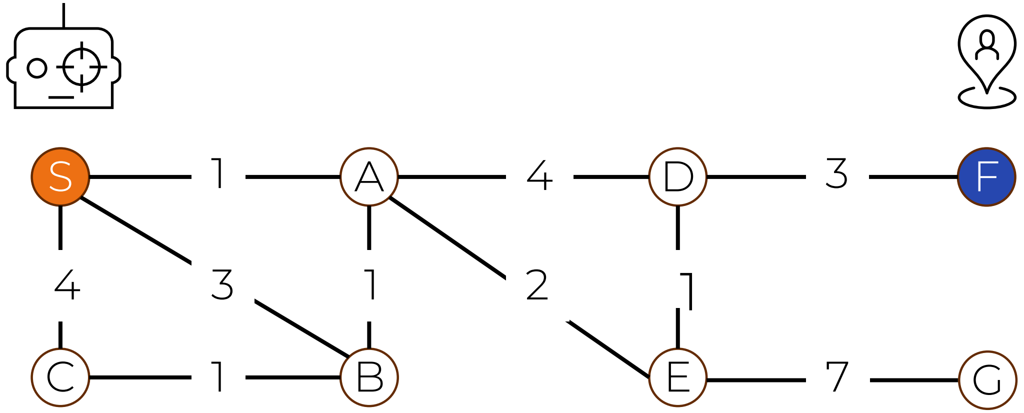
$S \xrightarrow{4} C$

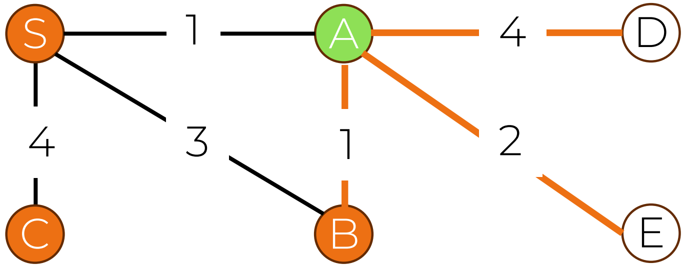
Мы знаем кратчайший путь из S в A

$S \xrightarrow{1} A$

Продолжаем исследовать граф в направлении А

ШАГ 2: ИССЛЕДУЕМ ВЕРШИНУ А





Из S можно добраться в A, B, C, D, E

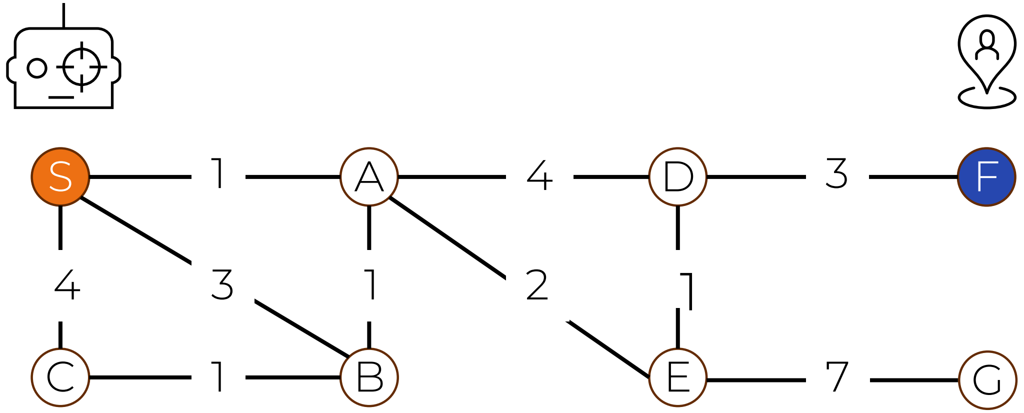
$S \xrightarrow{1} A$ $S \xrightarrow{5} D$
$S \xrightarrow{2} B$ $S \xrightarrow{3} E$
$S \xrightarrow{4} C$

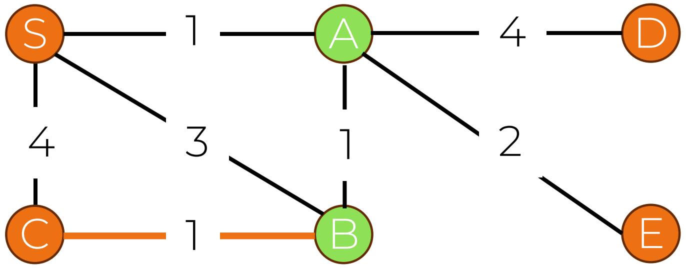
Мы знаем кратчайший путь из S в A и B

$S \xrightarrow{1} A$
$S \xrightarrow{2} B$

Продолжаем исследовать граф в направлении B

ШАГ 3: ИССЛЕДУЕМ ВЕРШИНУ B





Из S можно добраться в A, B, C, D, E

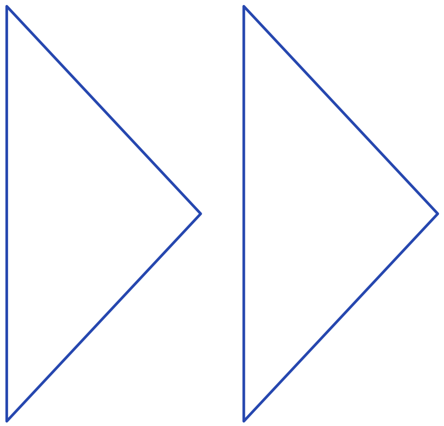
$S \to^{1} A$ $S \to^{5} D$
$S \to^{2} B$ $S \to^{3} E$
$S \to^{3} C$

Мы знаем кратчайший путь из S в A, B, C, E

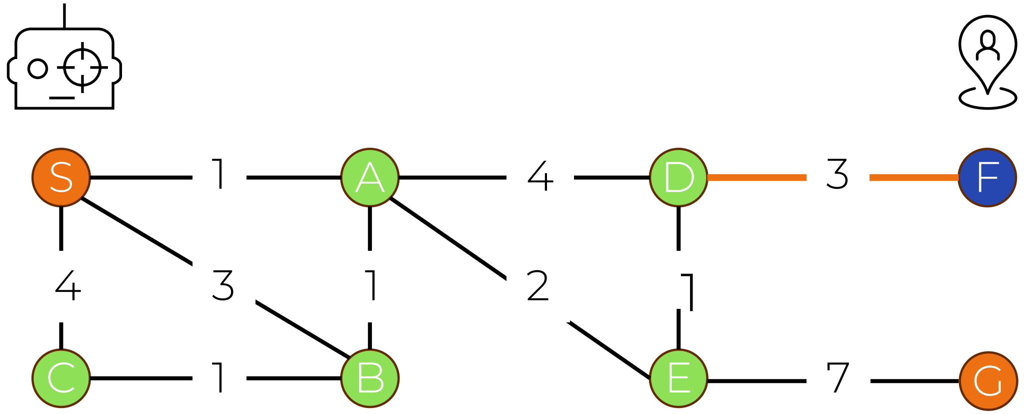
$S \to^{1} A$ $S \to^{3} E$
$S \to^{2} B$
$S \to^{3} C$

Продолжаем исследовать граф в направлении E[1]

[1]Выбор между E и C осуществляется произвольно



ФИНАЛЬНЫЙ ШАГ



Из S можно добраться в A, B, C, D, E, F, G

$S \xrightarrow{1} A$ $S \xrightarrow{4} D$
$S \xrightarrow{2} B$ $S \xrightarrow{3} E$
$S \xrightarrow{3} C$ $S \xrightarrow{7} F$
$S \xrightarrow{10} G$

Мы знаем кратчайший путь из S в A, B, C, D, E, F

$S \xrightarrow{1} A$ $S \xrightarrow{4} D$
$S \xrightarrow{2} B$ $S \xrightarrow{3} E$
$S \xrightarrow{3} C$ $S \xrightarrow{7} F$

Мы знаем кратчайший путь из S в F

АЛГОРИТМ: UNINFORMED SEARCH

```
FRONTIER NODES = S

EXPANDED NODES = { }

Пока цель F не попала в EXPANDED NODES:
```

- Найти вершину X с минимальным расстоянием от S в FRONTIER NODES

$$X = \underset{x \in FN}{\operatorname{argmin}}\ g(x)$$

- Исследовать (EXPAND) вершину X

- Добавить X в EXPANDED NODES и удалить ее из FRONTIER NODES

\# вершины, до которых мы знаем путь из S

\# вершины, до которых мы знаем **кратчайший** путь из S

\# g(x) – длина текущего пути из S в x

\# добавляем соседей X в FRONTIER NODES (если они еще не в EXPANDED NODES) и обновляем длины текущих путей

ТЕОРЕМА

АЛГОРИТМ UNINFORMED SEARCH ГАРАНТИРОВАННО НАЙДЕТ КРАТЧАЙШИЙ ПУТЬ ИЗ S В F

РАБОТА АЛГОРИТМА НА НАШЕМ ГРАФЕ

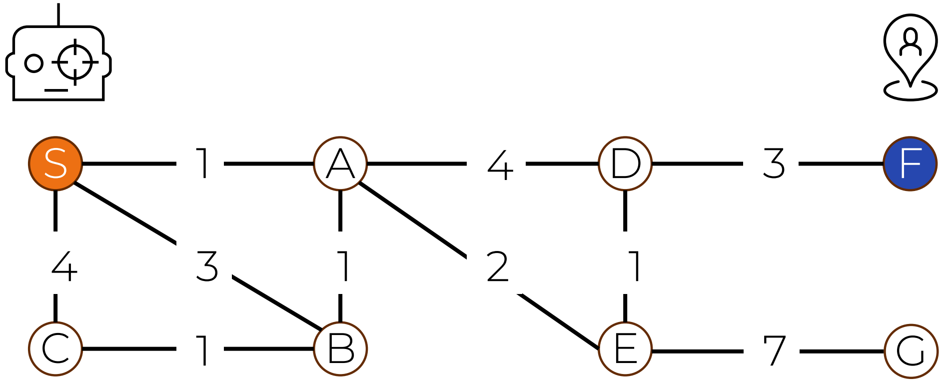
– оптимальный путь

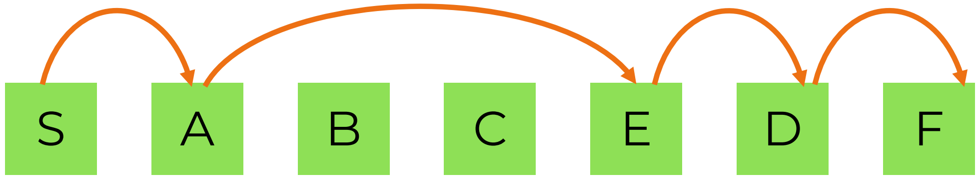
```
FRONTIER NODES = S

EXPANDED NODES = { }

Пока цель F не попала в EXPANDED NODES:
```

- Найти вершину X с минимальным расстоянием от S в FRONTIER NODES

$$X = \underset{x \in FN}{\mathrm{argmin}}\ g(x)$$

- Исследовать (EXPAND) вершину X
- Добавить X в EXPANDED NODES и удалить ее из FRONTIER NODES





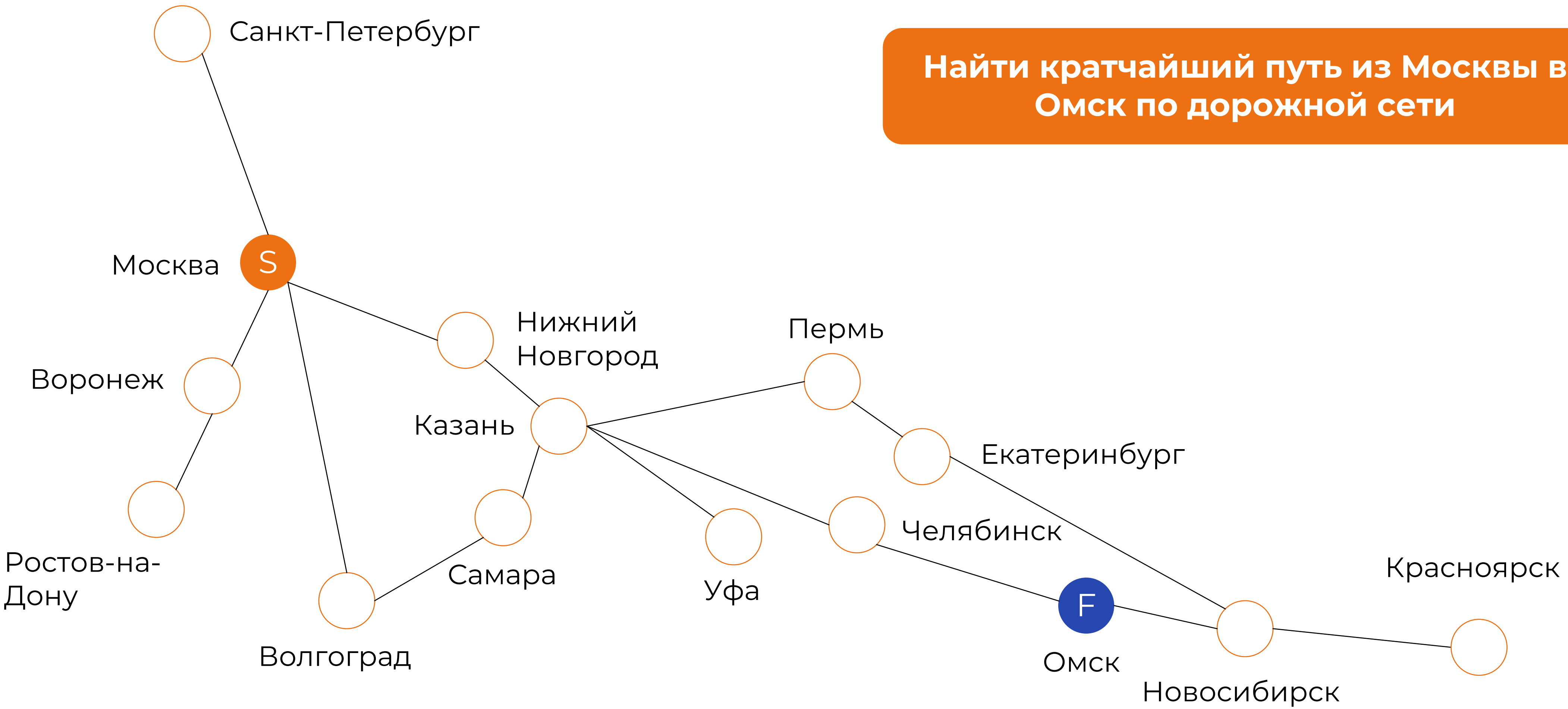
Порядок исследования вершин

ЧТО МОЖНО УЛУЧШИТЬ?

МОТИВАЦИЯ: GOOGLE MAPS

— – автомобильная дорога

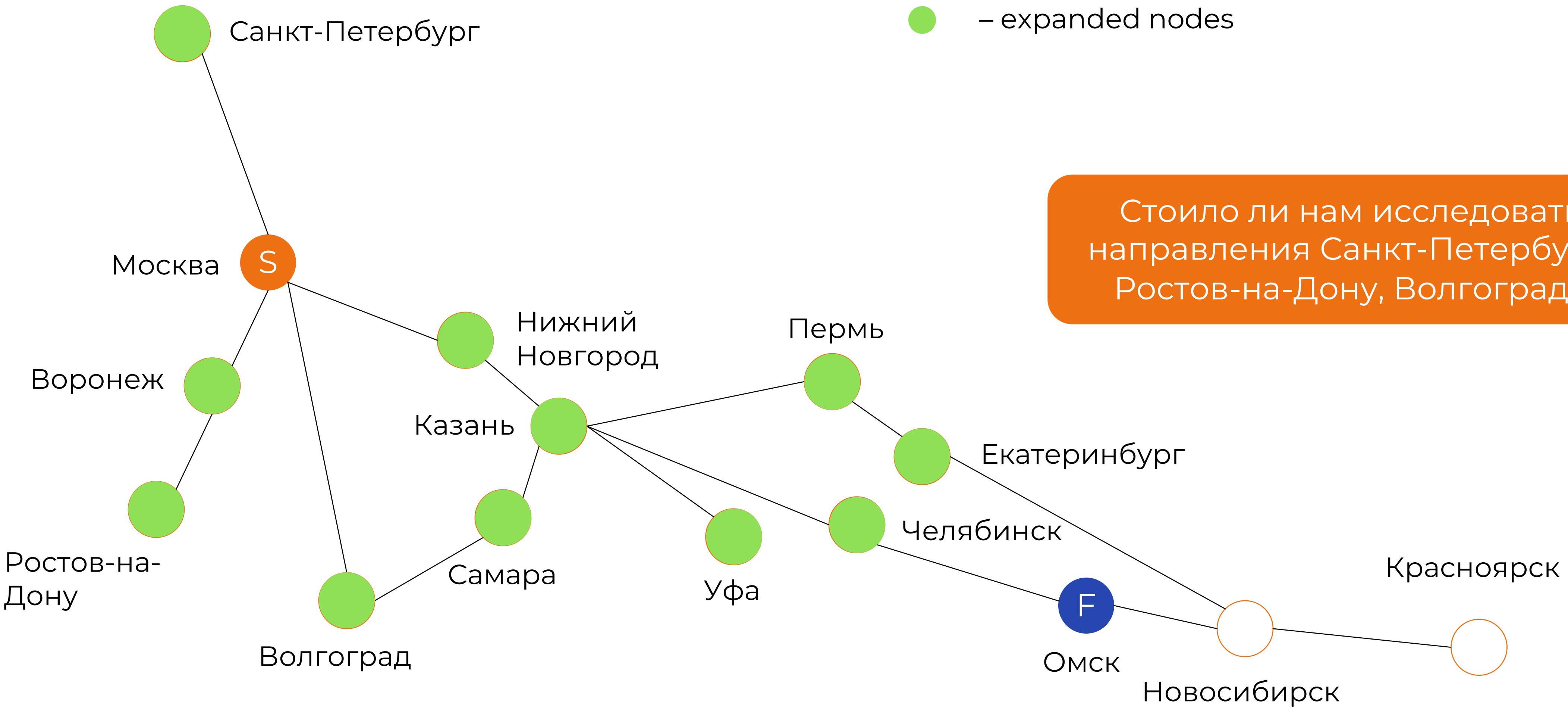
Найти кратчайший путь из Москвы в Омск по дорожной сети



МОТИВАЦИЯ: GOOGLE MAPS

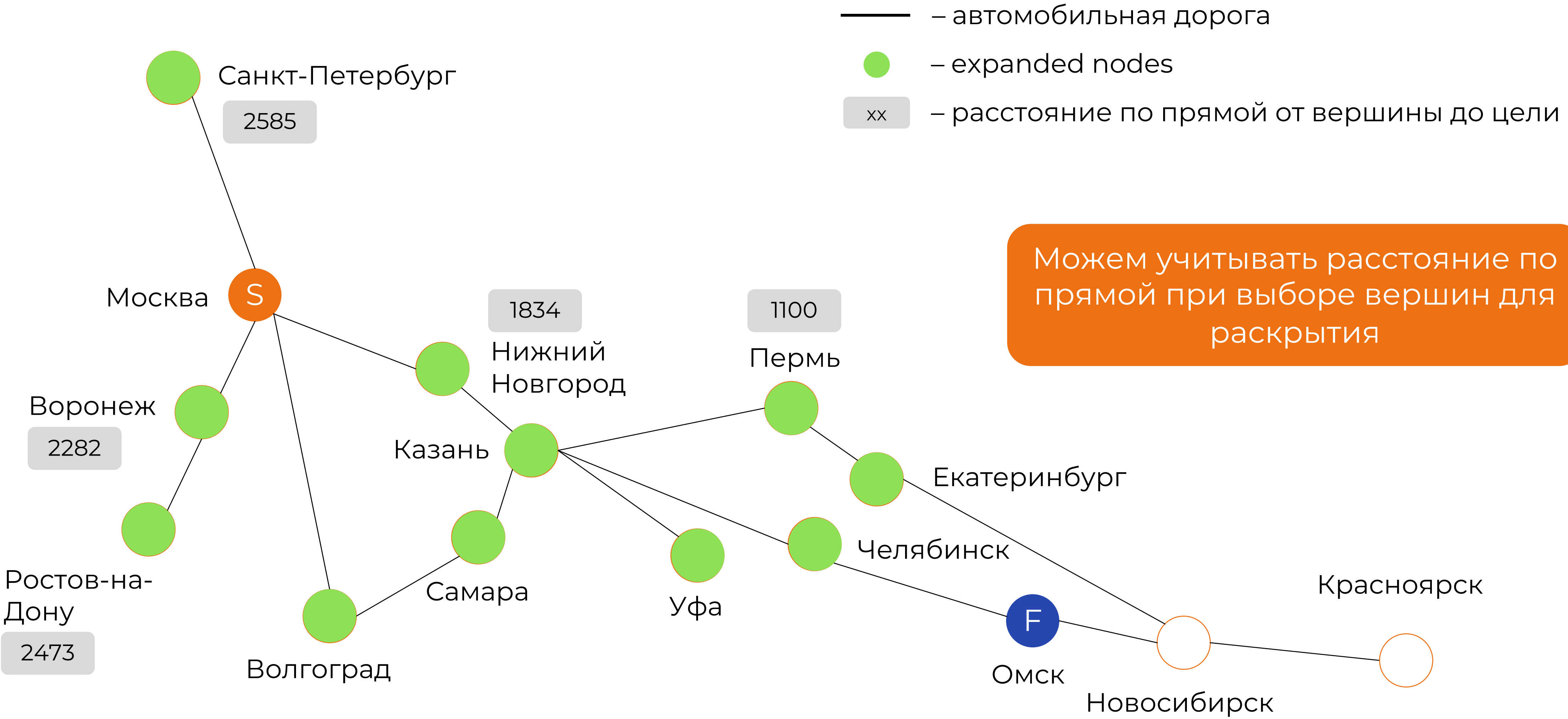
– автомобильная дорога

– expanded nodes



Стоило ли нам исследовать направления Санкт-Петербург, Ростов-на-Дону, Волгоград?

МОТИВАЦИЯ: GOOGLE MAPS



Можем учитывать расстояние по прямой при выборе вершин для раскрытия

Informed search

Если у нас есть способ **оценить длину оптимального пути** из каждой вершины до цели, мы можем ускорить работу алгоритма, раскрывая меньше вершин.

ОПРЕДЕЛЕНИЕ

ЭВРИСТРИКА

Функция h(X), которая для любой вершины X дает оценку длины оптимального пути из X в F

АЛГОРИТМ: A* SEARCH

UNINFORMED SEARCH

```
FRONTIER NODES = S

EXPANDED NODES = { }

Пока цель F не попала в EXPANDED NODES:
```

- Найти вершину X с минимальным расстоянием от S в FRONTIER NODES

$$X = \underset{x \in FN}{\mathrm{argmin}}\ g(x)$$

- Исследовать (EXPAND) вершину X
- Добавить X в EXPANDED NODES и удалить ее из FRONTIER NODES

g(x) – длина текущего пути из S в x

A* SEARCH

```
FRONTIER NODES = S

EXPANDED NODES = { }

Пока цель F не попала в EXPANDED NODES:
```

- Найти вершину X из FRONTIER NODES по критерию

$$X = \underset{x \in FN}{\mathrm{argmin}}\ (g(x) + h(x))$$

- Исследовать (EXPAND) вершину X
- Добавить X в EXPANDED NODES и удалить ее из FRONTIER NODES

h(x) – значение эвристики в вершине x

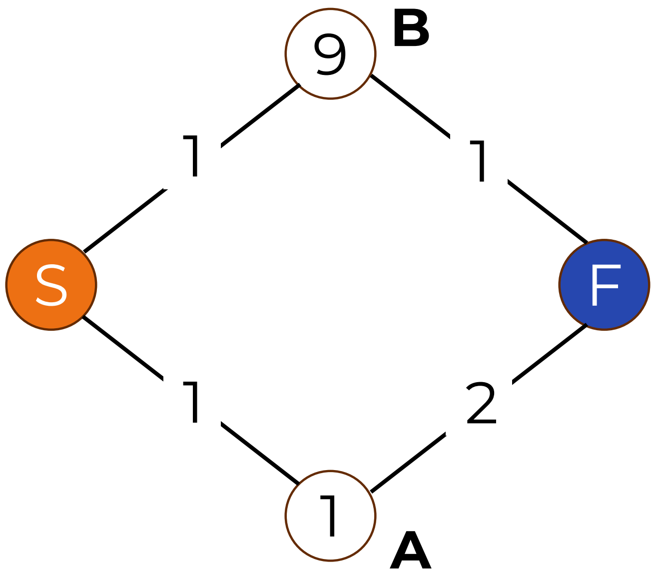
ОПТИМАЛЬНОСТЬ A* SEARCH

ГАРАНТИРУЕТ ЛИ A* НАХОЖДЕНИЕ ОПТИМАЛЬНОГО ПУТИ?

КОНТРПРИМЕРЫ

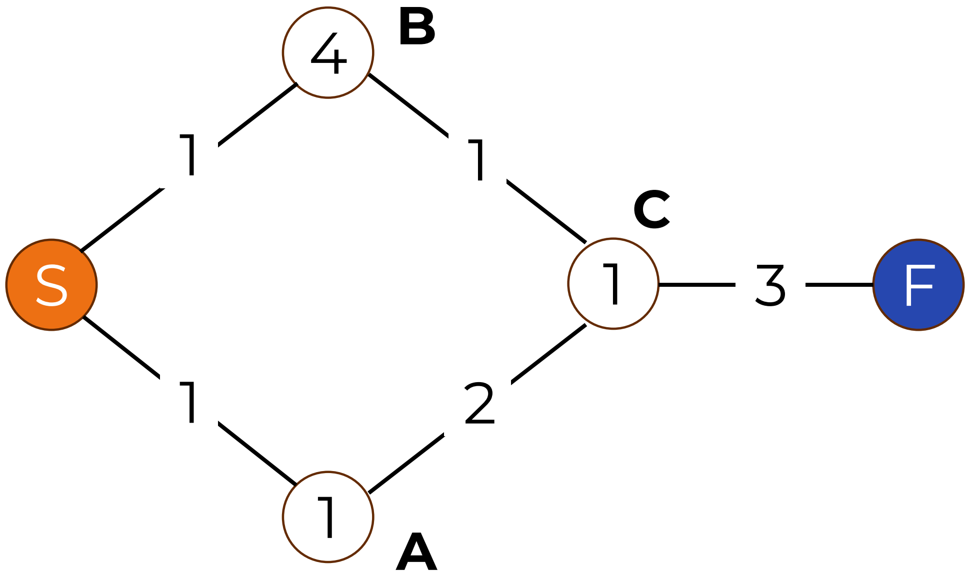
x – значение эвристики

ПРИМЕР 1: ПЕССИМИСТИЧНАЯ ЭВРИСТИКА



h(B) переоценивает длину оптимального пути

ПРИМЕР 2: НЕРАВЕНСТВО ТРЕУГОЛЬНИКА



h(B) > 2 + h(A)

ОПРЕДЕЛЕНИЕ

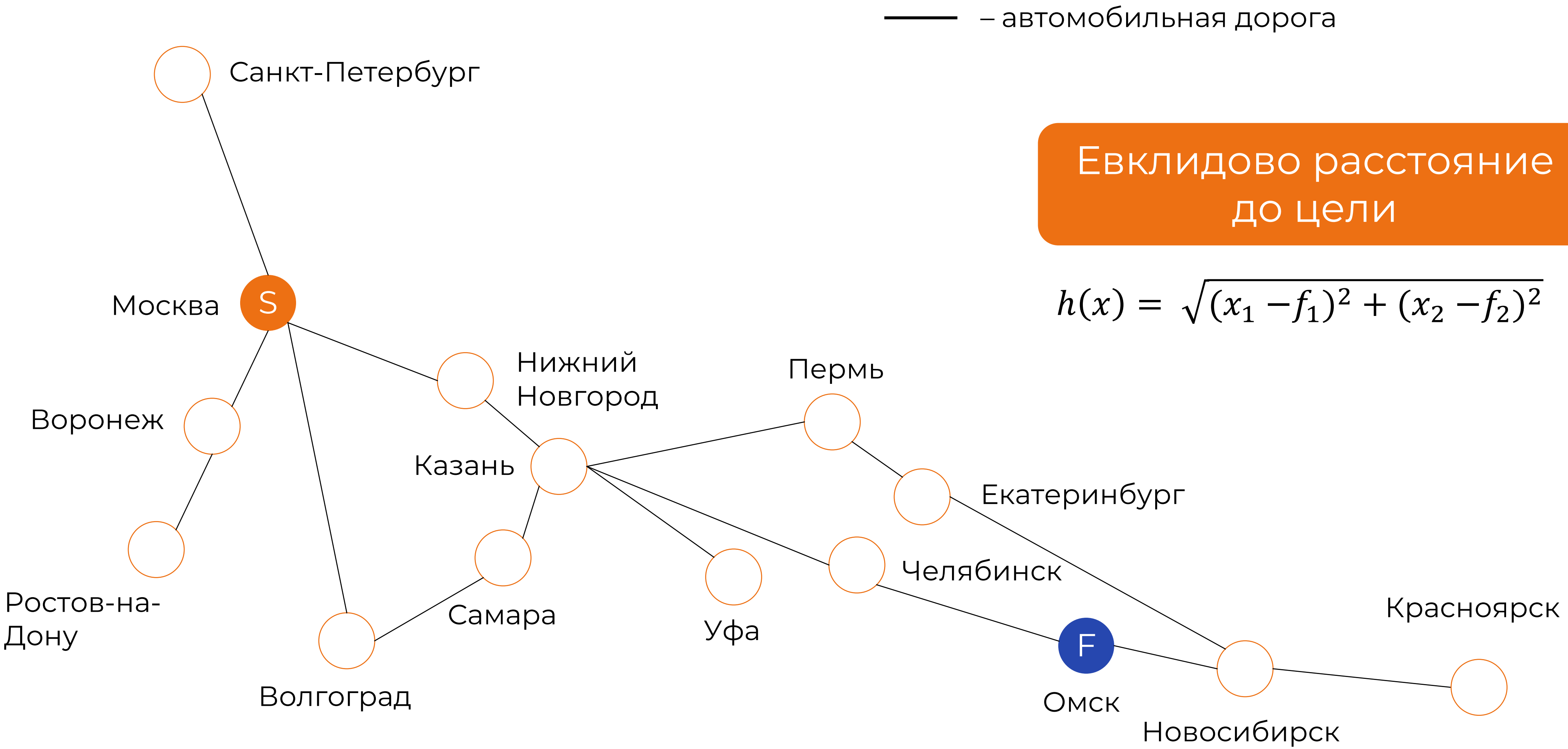
КОНСИСТЕНТНАЯ ЭВРИСТРИКА

Эвристика $h(X)$ называется консистентной если для любых двух вершин X и Y выполняется:

$$h(X) \leq c(X, Y) + h(Y)$$

c(X, Y) – длина кратчайшего пути из X в Y в графе

КОНСИСТЕНТНАЯ ЭВРИСТИКА: ПРИМЕР 1

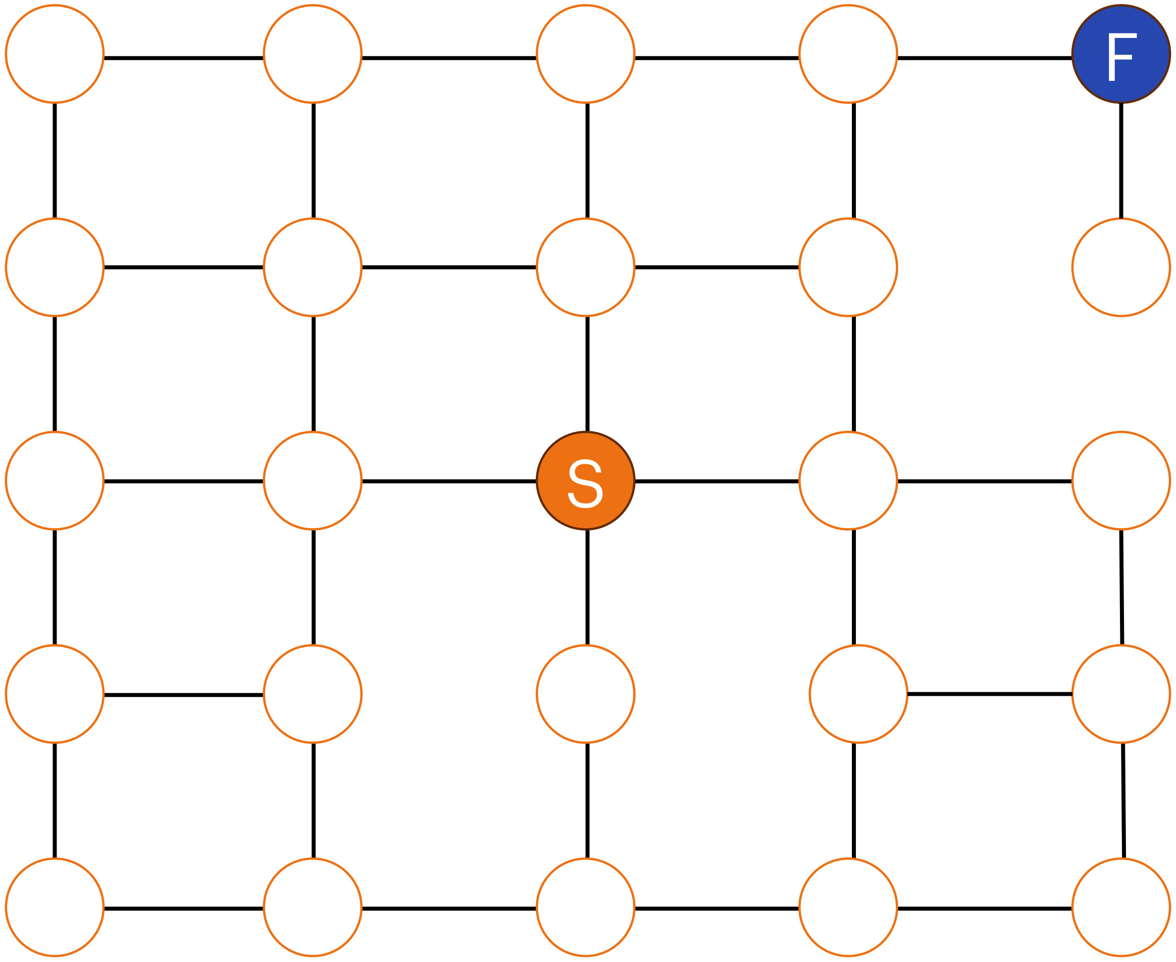


Евклидово расстояние до цели

$$h(x) = \sqrt{(x_1 - f_1)^2 + (x_2 - f_2)^2}$$

ПРИМЕР 2: КОНСИСТЕНТНАЯ ЭВРИСТИКА

– автомобильная дорога длины 1



Евклидово расстояние до цели

$$h(x) = \sqrt{(x_1 - f_1)^2 + (x_2 - f_2)^2}$$

Manhattan distance

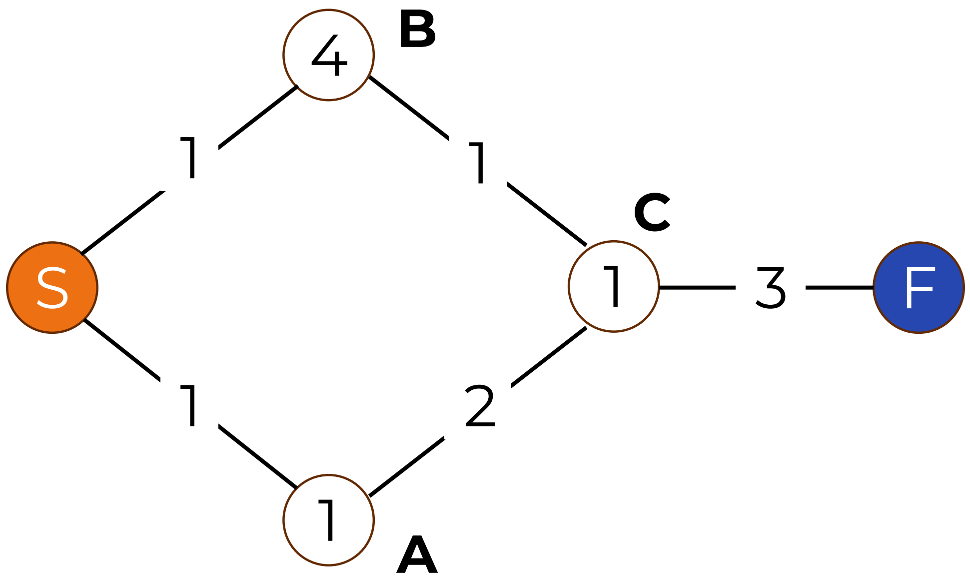
$$h(x) = |x_1 - f_1| + |x_2 - f_2|$$

ТЕОРЕМА

АЛГОРИТМ A* SEARCH С КОНСИСТЕНТНОЙ ЭВРИСТИКОЙ ГАРАНТИРОВАННО НАЙДЕТ КРАТЧАЙШИЙ ПУТЬ ИЗ S В F

НЕКОНСИСТЕНТНАЯ ЭВРИСТИКА

x – значение эвристики



Можем ли мы применить идеи A* к этому случаю?

АЛГОРИТМ: A* TREE SEARCH

A* SEARCH

```
FRONTIER NODES = S

EXPANDED NODES = { }

Пока цель F не попала в EXPANDED NODES:

■ Найти вершину X из FRONTIER NODES
  по критерию

          X = argmin g(x) + h(x)
                x ∈ FN

■ Исследовать (EXPAND) вершину X

■ Добавить X в EXPANDED NODES и
  удалить ее из FRONTIER NODES
```

A* search
добавляем соседей X в FRONTIER NODES (если они еще не в EXPANDED NODES) и обновляем длины текущих путей

A* tree search
добавляем соседей X в FRONTIER NODES ~~(если они еще не в EXPANDED NODES)~~ и обновляем длины текущих путей

\# g(x) – длина текущего пути из S в x

\# h(x) – значение эвристики в вершине x

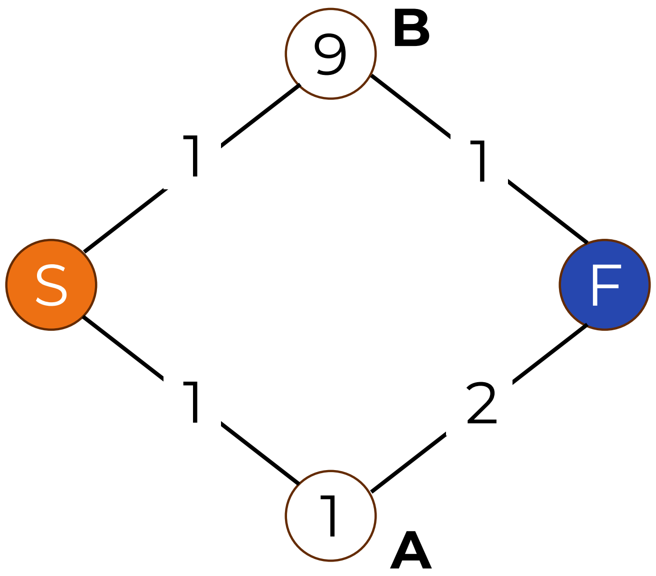
ОПТИМАЛЬНОСТЬ A* TREE SEARCH

ГАРАНТИРУЕТ ЛИ A* TREE SEARCH НАХОЖДЕНИЕ ОПТИМАЛЬНОГО ПУТИ?

КОНТРПРИМЕР

x – значение эвристики

ПРИМЕР 1: ПЕССИМИСТИЧНАЯ ЭВРИСТИКА	ИНСАЙТ



h(B) переоценивает длину оптимального пути

Нужно быть оптимистичным!

ОПРЕДЕЛЕНИЕ

ДОПУСТИМАЯ ЭВРИСТРИКА

Эвристика $h(X)$ называется допустимой если для любой вершины X:

$$h(X) \leq c(X, F)$$

c(X, Y) – длина кратчайшего пути из X в Y в графе

ОТНОШЕНИЕ МЕЖДУ ДОПУСТИМОЙ И КОНСИСТЕНТНОЙ ЭВРИСТИКОЙ?

ТЕОРЕМА

АЛГОРИТМ A* TREE SEARCH С ДОПУСТИМОЙ ЭВРИСТИКОЙ ГАРАНТИРОВАННО НАЙДЕТ КРАТЧАЙШИЙ ПУТЬ ИЗ S В F

ПОИСК ЭТО ТОЛЬКО ПРО МАРШРУТЫ?

Discrete Mathematics 27 (1979) 47–57.

BOUNDS FOR SORTING BY PREFIX REVERSAL

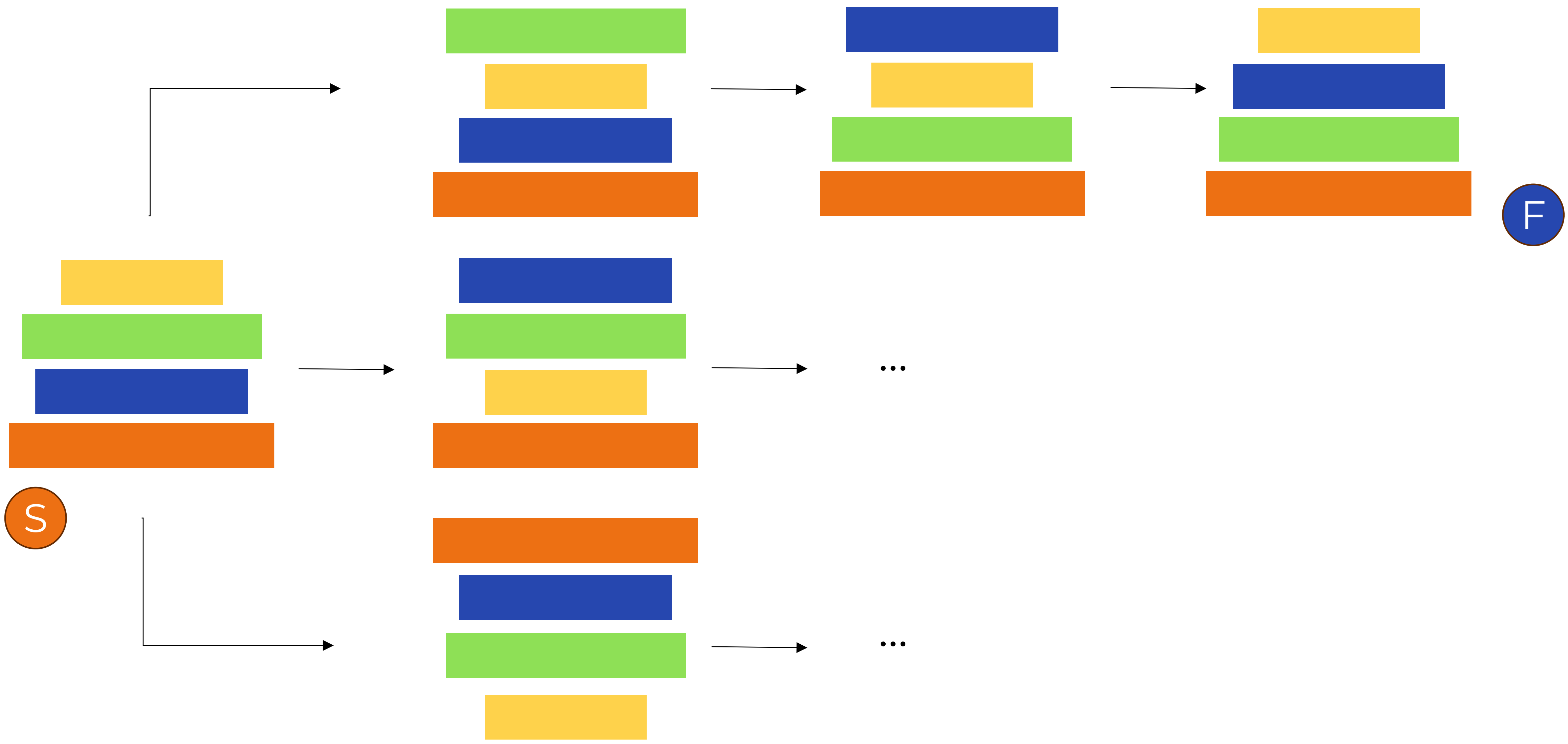
William H. GATES
Microsoft, Albuquerque, New Mexico

Christos H. PAPADIMITRIOU*†
Department of Electrical Engineering, University of California, Berkeley, CA 94720, U.S.A.

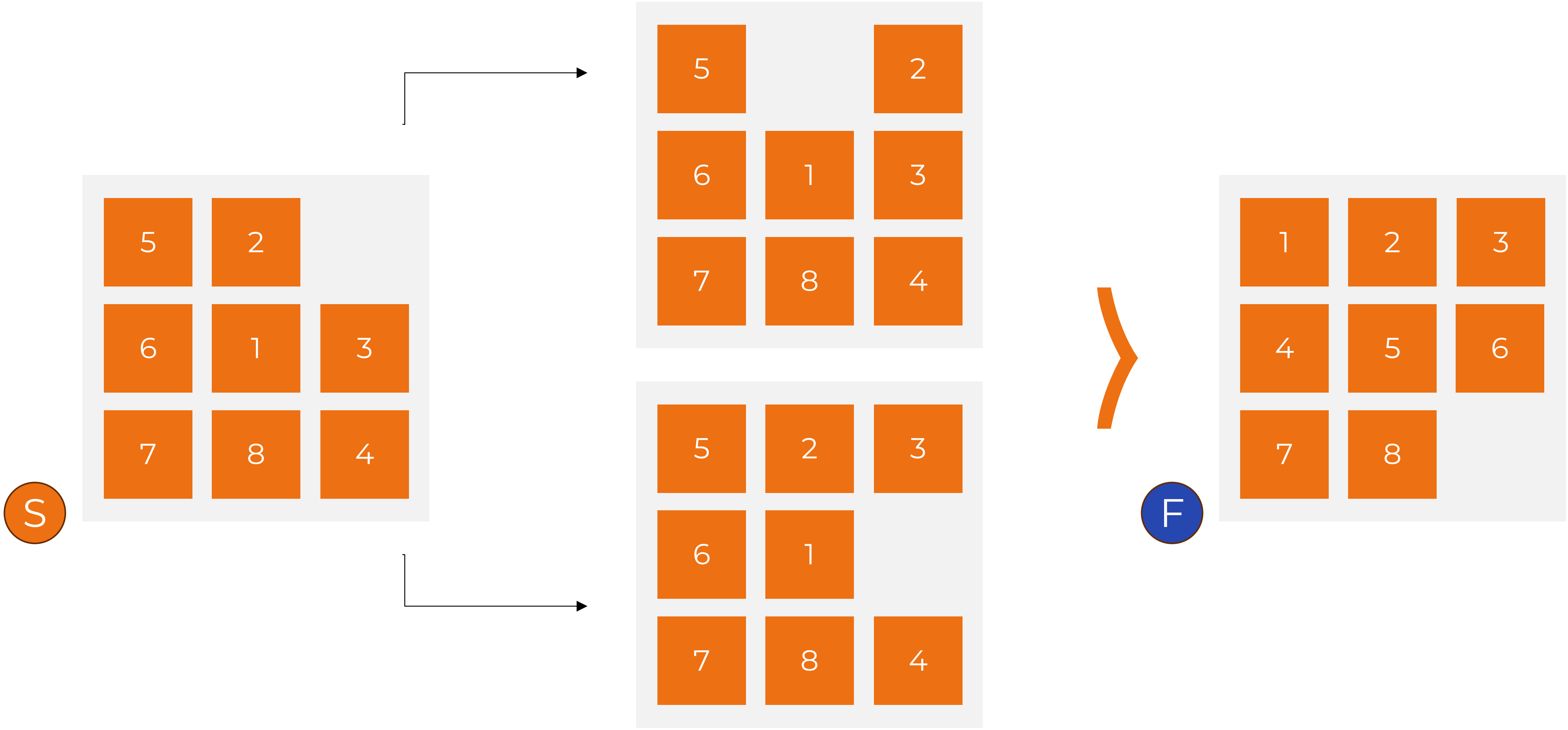
Received 18 January 1978
Revised 28 August 1978

For a permutation σ of the integers from 1 to n, let $f(\sigma)$ be the smallest number of prefix reversals that will transform σ to the identity permutation, and let $f(n)$ be the largest such $f(\sigma)$ for all σ in (the symmetric group) S_n. We show that $f(n) \leqslant (5n+5)/3$, and that $f(n) \geqslant 17n/16$ for n a multiple of 16. If, furthermore, each integer is required to participate in an even number of reversed prefixes, the corresponding function $g(n)$ is shown to obey $3n/2-1 \leqslant g(n) \leqslant 2n+3$.

ПРИМЕР: PANCAKES



ПРИМЕР: ПЯТНАШКИ



ОБЩИЙ ВИД ЗАДАЧИ ПОИСКА

Алгоритмы поиска можно применять для решения широкого класса задач

Состояния
Конфигурации задачи
→ **Вершины графа**

Действия
Возможные действия в каждом состоянии
→ **Ребра графа**

Стоимость
Цена каждого действия
→ **Веса ребер**

САММАРИ

ИНСАЙТ
Нужно быть оптимистичным!