

Лабораторная работа 4


Тема: Исследование поиска с ограничением глубины.

Цель работы: приобретение навыков по работе с поиском с ограничением глубины с помощью языка программирования Python версии 3.x

Ссылка на GitHub: https://github.com/RomanGorchakov/AI_4

Порядок выполнения работы

1. Создаём новый общедоступный репозиторий в Github, в котором будет использована лицензия MIT и язык программирования Python.

 Start writing code ...

Start a new repository for RomanGorchakov

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name *

AI_4

✓ AI_4 is available.

☒ **Public**
Anyone on the internet can see this repository

☐ **Private**
You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

RomanGorchakov / README.md

Create

```
1 - 🙋 Hi, I'm @RomanGorchakov
2 - 👀 I'm interested in ...
3 - 🌱 I'm currently learning ...
4 - ❤️ I'm looking to collaborate on ...
5 - 💻 How to reach me ...
6 - 🗨️ Pronouns: ...
7 - ⚡ Fun fact: ...
8
```

RomanGorchakov / AI_4

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Add a license to your project

License	Permissions	Limitations	Conditions
Apache License 2.0			
GNU General Public License v3.0			
MIT License	<ul style="list-style-type: none"> ✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use 	<ul style="list-style-type: none"> ✗ Liability ✗ Warranty 	<ul style="list-style-type: none"> 🔗 License and copyright notice
BSD 2-Clause "Simplified" License			
BSD 3-Clause "New" or "Revised" License			
Boost Software License 1.0			
Creative Commons Zero v1.0 Universal			
Eclipse Public License 2.0			
GNU Affero General Public License			

This is not legal advice. [Learn more about repository licenses.](#)

MIT License

Copyright (c) year full name

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge,

To adopt MIT License, enter your details. You'll have a chance to review before committing a LICENSE file to a new branch or the root of your project.

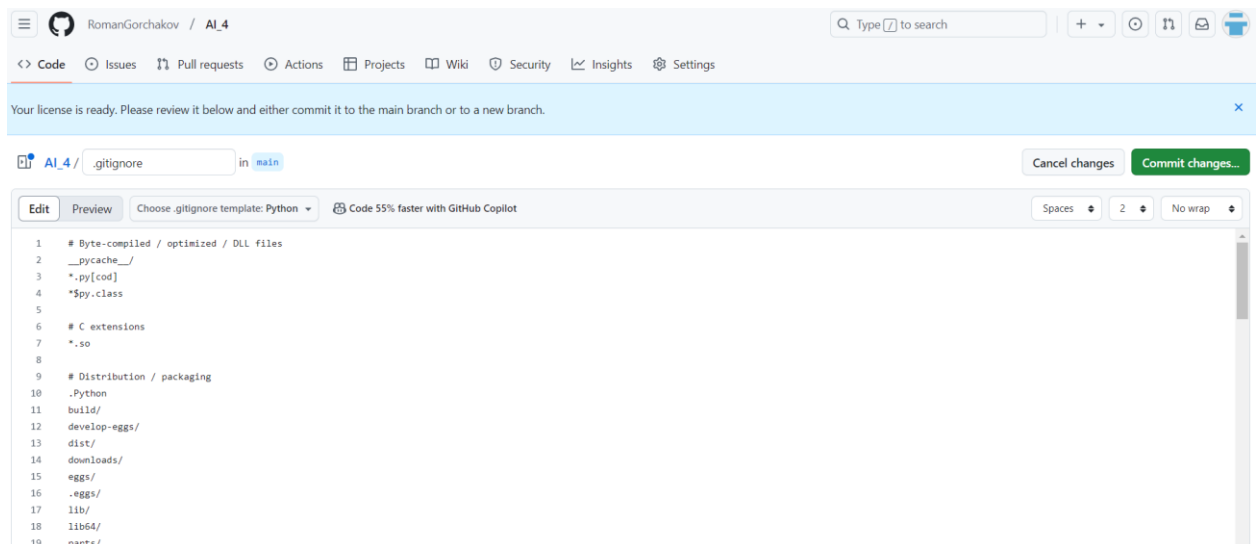
Year ^①
2024

Full name ^①
RomanGorchakov

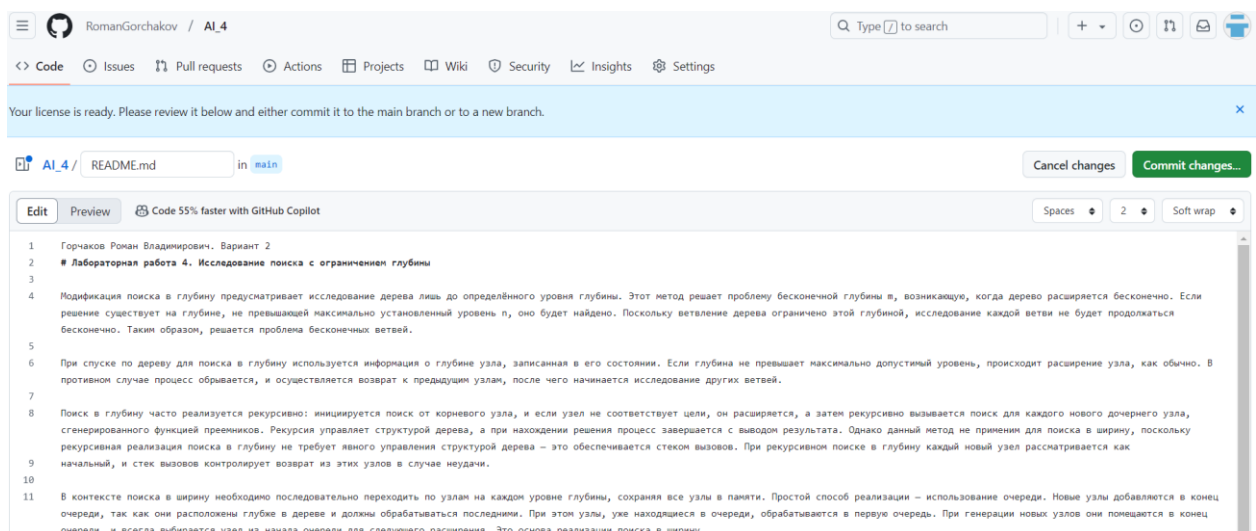
[Review and submit](#)

2. Теперь необходимо дополнить файл `.gitignore` с необходимыми правилами для языка программирования Python. Для этого переходим по ссылке «<https://github.com/github/gitignore>» и скачиваем оттуда файл «Python.gitignore».

Phalcon.gitignore	Remove trailing asterisks in Phalcon rules	10 years ago
PlayFramework.gitignore	Added /project/project to PlayFramework.gitignore	7 years ago
Plone.gitignore	Covered by global vim template	10 years ago
Prestashop.gitignore	Update for Prestashop 1.7 (#3261)	3 years ago
Processing.gitignore	Ignore transpiled .java and .class files (#3016)	4 years ago
PureScript.gitignore	Update PureScript adding .spago (#3278)	3 years ago
Python.gitignore	Update Python.gitignore	last year
Qooxdoo.gitignore	Add gitignore for qooxdoo apps	13 years ago
Qt.gitignore	Remove trailing whitespace	2 years ago
R.gitignore	Merge pull request #3792 from jl5000/patch-1	2 years ago
README.md	Merge pull request #3854 from AnilSeervi/patch-1	2 years ago
ROS.gitignore	Added ignore for files created by <code>catkin_make_isolated</code>	6 years ago
Racket.gitignore	Update Racket.gitignore	2 years ago
Rails.gitignore	Ignore Rails .env according recommendations	2 years ago
Raku.gitignore	Changes the name of Perl 6 to Raku (#3312)	3 years ago
RhodesRhomobile.gitignore	Add Rhodes mobile application framework gitignore	13 years ago
Ruby.gitignore	Ruby: ignore RuboCop remote inherited config files (#3197)	4 years ago



3. Теперь создаём файл «README.md», где вносим ФИО и теоретический конспект лекции. Сохраняем набранный текст через кнопку «Commit changes».



4. В окне «Codespace» выбираем опцию «Create codespace on main». Откроется терминал, куда мы введём команду «git clone», чтобы клонировать свой репозиторий. После этого организуем репозиторий в соответствии с моделью ветвления Git-flow. Для этого введём в терминал команды: «git checkout –b develop» для создания ветки разработки; «git branch feature_branch» для создания ветки функций; «git branch release/1.0.0» для создания ветки релиза; «git checkout main» и «git branch hotfix» для создания веток hotfix. Устанавливаем библиотеки isort, black и flake8 и создаём файлы .pre-commit-config.yaml и environment.yml.

```

@RomanGorchakov →/workspaces/AI_4 (main) $ git checkout -b develop
Switched to a new branch 'develop'
• @RomanGorchakov →/workspaces/AI_4 (develop) $ git branch feature_branch
• @RomanGorchakov →/workspaces/AI_4 (develop) $ git branch release/1.0.0
• @RomanGorchakov →/workspaces/AI_4 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @RomanGorchakov →/workspaces/AI_4 (main) $ git branch hotfix
• @RomanGorchakov →/workspaces/AI_4 (main) $ git checkout develop
Switched to branch 'develop'
○ @RomanGorchakov →/workspaces/AI_4 (develop) $ █

```

```

Collecting black
  Downloading black-24.10.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl.metadata (79 kB)
Collecting click>=8.0.0 (from black)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting mypy-extensions>=0.4.3 (from black)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: packaging>=22.0 in /home/codespace/.local/lib/python3.12/site-packages (from black) (24.1)
Collecting pathspec>=0.9.0 (from black)
  Downloading pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: platformdirs>=2 in /home/codespace/.local/lib/python3.12/site-packages (from black) (4.3.6)
Downloading black-24.10.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_28_x86_64.whl (1.8 MB)
----- 1.8/1.8 MB 11.4 MB/s eta 0:00:00
Downloading click-8.1.7-py3-none-any.whl (97 kB)
Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Downloading pathspec-0.12.1-py3-none-any.whl (31 kB)
Installing collected packages: pathspec, mypy-extensions, click, black
Successfully installed black-24.10.0 click-8.1.7 mypy-extensions-1.0.0 pathspec-0.12.1
• @RomanGorchakov →/workspaces/AI_4 (develop) $ pip install flake8
Collecting flake8
  Downloading flake8-7.1.1-py2.py3-none-any.whl.metadata (3.8 kB)
Collecting mccabe<0.8.0,>=0.7.0 (from flake8)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Collecting pycodestyle<2.13.0,>=2.12.0 (from flake8)
  Downloading pycodestyle-2.12.1-py2.py3-none-any.whl.metadata (4.5 kB)
Collecting pyflakes<3.3.0,>=3.2.0 (from flake8)
  Downloading pyflakes-3.2.0-py2.py3-none-any.whl.metadata (3.5 kB)
Downloading flake8-7.1.1-py2.py3-none-any.whl (57 kB)
Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Downloading pycodestyle-2.12.1-py2.py3-none-any.whl (31 kB)
Downloading pyflakes-3.2.0-py2.py3-none-any.whl (62 kB)
Installing collected packages: pyflakes, pycodestyle, mccabe, flake8
Successfully installed flake8-7.1.1 mccabe-0.7.0 pycodestyle-2.12.1 pyflakes-3.2.0
• @RomanGorchakov →/workspaces/AI_4 (develop) $ pre-commit sample-config > .pre-commit-config.yaml
• @RomanGorchakov →/workspaces/AI_4 (develop) $ conda env export > environment.yml
○ @RomanGorchakov →/workspaces/AI_4 (develop) $ █

```

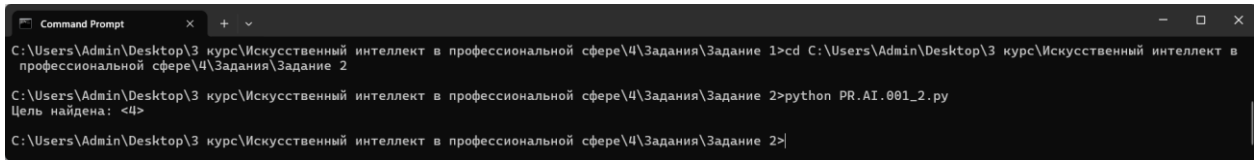
5. Создаём файл «PR.AI.001_1.py», в котором нужно разработать систему навигации для робота-пылесоса.

```

Command Prompt
C:\Users\Admin>cd C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 1
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 1>python PR.AI.001_1.py
Найден на глубине: True
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 1>

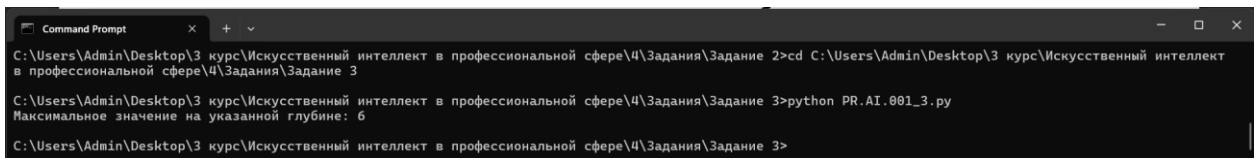
```

6. Создаём файл «PR.AI.001_2.py», в котором нужно разработать систему управления складом.



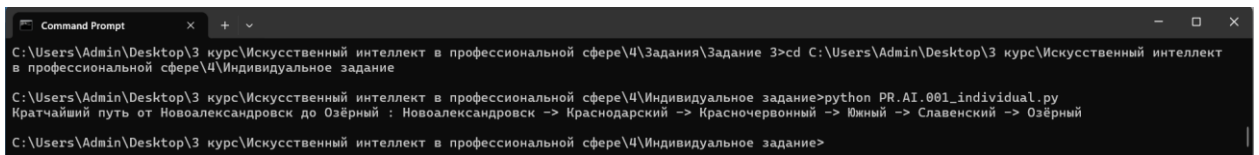
```
Command Prompt
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 1>cd C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 2
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 2>python PR.AI.001_2.py
Цель найдена: <4>
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 2>
```

7. Создаём файл «PR.AI.001_3.py», в котором нужно разработать систему автоматического управления инвестициями.



```
Command Prompt
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 2>cd C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 3
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 3>python PR.AI.001_3.py
Максимальное значение на указанной глубине: 6
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 3>
```

8. Создаём файл «PR.AI.001_individual.py», в котором нужно найти минимальное расстояние между начальным и конечным пунктами для построенного графа лабораторной работы 1 с использованием алгоритма поиска с ограничением глубины.



```
Command Prompt
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Задания\Задание 3>cd C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Индивидуальное задание
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Индивидуальное задание>python PR.AI.001_individual.py
Кратчайший путь от Новоалександровск до Озёрный : Новоалександровск -> Краснодарский -> Краснотеревонный -> Южный -> Славенский -> Озёрный
C:\Users\Admin\Desktop\3 курс\Искусственный интеллект в профессиональной сфере\4\Индивидуальное задание>
```

9. Выполняем коммит файлов в репозиторий Git в ветку разработки, сливаем её с веткой main и отправляем изменения на сервер GitHub.

AI_4

Public

Pin

Unwatch 1

Fork 0

Star 0

main

1 Branch

0 Tags

Go to file

Add file

Code

RomanGorchakov

Depth Limit Search

a6d5260 · now

4 Commits

Задания	Depth Limit Search	now
Индивидуальное задание	Depth Limit Search	now
Отчёт	Depth Limit Search	now
.gitignore	Create .gitignore	6 hours ago
.pre-commit-config.yaml	Depth Limit Search	now
.python-version	Depth Limit Search	now
LICENSE	Create LICENSE	6 hours ago
README.md	Create README.md	3 hours ago
edu.pyoop.code-workspace	Depth Limit Search	now
environment.yml	Depth Limit Search	now
pyproject.toml	Depth Limit Search	now
setup.cfg	Depth Limit Search	now
uv.lock	Depth Limit Search	now

README

MIT license

Горчаков Роман Владимирович. Вариант 2

Лабораторная работа 4. Исследование поиска с ограничением глубины

About

No description, website, or topics provided.

Readme

MIT license

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

Python 100.0%

Suggested workflows

Based on your tech stack

Python application

Configure

Create and test a Python application.

SLSA Generic generator

Configure

Generate SLSA3 provenance for your existing release workflows

Контрольные вопросы

1. Что такое поиск с ограничением глубины, и как он решает проблему бесконечных ветвей?

Модификация поиска в глубину предусматривает исследование дерева лишь до определённого уровня глубины. Если решение существует на глубине, не превышающей максимально установленный уровень n , оно будет найдено. Поскольку ветвление дерева ограничено этой глубиной, исследование каждой ветви не будет продолжаться бесконечно.

2. Какова основная цель ограничения глубины в данном методе поиска?

Этот метод решает проблему бесконечной глубины m , возникающую, когда дерево расширяется бесконечно.

3. В чем разница между поиском в глубину и поиском с ограничением глубины?

Поиск в глубину – стратегия поиска решений в пространстве состояний, при которой всегда развёртывается самый глубокий узел в текущей периферии дерева поиска.

Поиск с ограничением глубины – это вариант поиска в глубину, в котором применяется заранее определённый предел глубины, что позволяет решить проблему бесконечного пути.

4. Какую роль играет проверка глубины узла в псевдокоде поиска с ограничением глубины?

Псевдокод указывает, что при спуске по дереву для поиска в глубину используется информация о глубине узла, записанная в его состоянии. Если глубина не превышает максимально допустимый уровень, происходит расширение узла, как обычно. В противном случае процесс обрывается, и осуществляется возврат к предыдущим узлам, после чего начинается исследование других ветвей.

5. Почему в случае достижения лимита глубины функция возвращает «обрезание»?

Потому что в этом случае останавливается рекурсия.

6. В каких случаях поиск с ограничением глубины может не найти решение, даже если оно существует?

Когда выбрано значение глубины, которое меньше или больше заранее заданной глубины.

7. Как поиск в ширину и в глубину отличаются при реализации с использованием очереди?

Простой способ реализации – использование очереди. Новые узлы добавляются в конец очереди, так как они расположены глубже в дереве и должны обрабатываться последними. При этом узлы, уже находящиеся в очереди, обрабатываются в первую очередь. При генерации новых узлов они помещаются в конец очереди, и всегда выбирается узел из начала очереди для следующего расширения. Это основа реализации поиска в ширину.

Поиск в глубину также может быть реализован с использованием очереди, при этом новые дочерние узлы помещаются в начало очереди. Это позволяет сначала исследовать самые глубокие ветви. Если поиск по дереву реализуется с помощью очереди, основное отличие между поиском в глубину и ширину заключается в порядке добавления новых узлов в очередь.

8. Почему поиск с ограничением глубины не является оптимальным?

Поиск с ограничением глубины не является оптимальным, потому что при большой глубине поиска затраты вычислительного времени могут быть существенными.

9. Как итеративное углубление улучшает стандартный поиск с ограничением глубины?

Итеративное углубление сочетает в себе лучшие качества поиска в глубину и ширину. Такой подход обеспечивает полноту и оптимальность

поиска в ширину, сохраняя при этом низкую пространственную сложность поиска в глубину.

10. В каких случаях итеративное углубление становится эффективнее простого поиска в ширину?

Итеративное углубление становится эффективнее простого поиска в ширину для глубоких деревьев. 1

Итеративное углубление эффективно для деревьев с высоким коэффициентом ветвления, так как оно избегает хранения большого числа узлов в памяти, что является проблемой для поиска в ширину. Каждая итерация увеличивает глубину, позволяя обнаруживать решения на меньших глубинах быстрее, чем поиск с фиксированной ограниченной глубиной, обеспечивая тем самым оптимальность результата.

11. Какова основная цель использования алгоритма поиска с ограничением глубины?

Повысить эффективность и обеспечить более управляемое время поиска.

12. Какие параметры принимает функция `depth_limited_search`, и каково их назначение?

Функция `depth_limited_search` принимает два аргумента: `problem` (задача, которую нужно решить) и `limit` (максимальная глубина поиска).

13. Какое значение по умолчанию имеет параметр `limit` в функции `depth_limited_search`?

В данном случае, по умолчанию `limit` равен 10.

14. Что представляет собой переменная `frontier`, и как она используется в алгоритме?

Переменная `frontier` представляет собой стек (LIFO-очередь), содержащий начальный узел. `Node(problem.initial)` создает начальный узел поиска, используя начальное состояние задачи `problem`.

15. Какую структуру данных представляет `LIFOQueue`, и почему она используется в этом алгоритме?

LIFOQueuee представляет собой структуру данных стека, которая работает по принципу «последний вошёл – первый вышел» (LIFO). Это означает, что элемент, добавленный последним, будет удалён первым.

LIFOQueuee используется в алгоритме поиска с ограничением глубины в комбинированном поиске, предложенном немецким математиком Корфом. В начале организуется традиционный поиск в глубину с использованием списка открытых вершин в виде LIFO. При достижении некоторого заданного уровня осуществляется изменение направления с глубины в ширину, и теперь список понимается как структура FIFO.

16. Каково значение переменной `result` при инициализации, и что оно означает?

Переменная `result` хранит результат поиска. Изначально она установлена в значение `failure`, что означает неудачу поиска.

17. Какое условие завершает цикл `while` в алгоритме поиска?

Цикл выполняется до тех пор, пока `frontier` не станет пустым, то есть пока есть узлы для рассмотрения.

18. Какой узел извлекается с помощью `frontier.pop()` и почему?

Извлекается последний добавленный узел из `frontier` для дальнейшей обработки.

19. Что происходит, если найден узел, удовлетворяющий условию цели (условие `problem.is_goal(node.state)`)?

Проверяется, является ли текущий узел целевым. Если да, то поиск завершен успешно.

20. Какую проверку выполняет условие `elif len(node) >= limit`, и что означает его выполнение?

Проверяется, достиг ли текущий узел ограничения по глубине. Если да, то дальнейший поиск в этом направлении прекращается.

21. Что произойдет, если текущий узел достигнет ограничения по глубине поиска?

Если текущий узел достиг ограничения по глубине, переменной `result` присваивается значение `cutoff`, что означает достижение лимита глубины поиска.

22. Какую роль выполняет проверка на циклы `elif not is_cycle(node)` в алгоритме?

Проверяется, не ведет ли текущий узел к циклу. Если нет, то можно продолжать поиск.

23. Что происходит с дочерними узлами, полученными с помощью функции `expand(problem, node)`?

Для каждого дочернего узла текущего узла (расширение текущего узла) выполняется итерация. Каждый дочерний узел добавляется в `frontier` для дальнейшей обработки. По завершению цикла возвращается результат поиска.

24. Какое значение возвращается функцией, если целевой узел не был найден?

Если целевой узел так и не был найден, возвращается значение `result`, которое может быть либо `failure`, либо `cutoff`, в зависимости от результата поиска.

25. В чем разница между результатами `failure` и `cutoff` в контексте данного алгоритма?

`failure` – неудача поиска. `cutoff` – достижение лимита глубины поиска.