In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [50]:

```
df = pd.read_table("processed_log1.md", sep=" ")
df.head()
```

Out[50]:

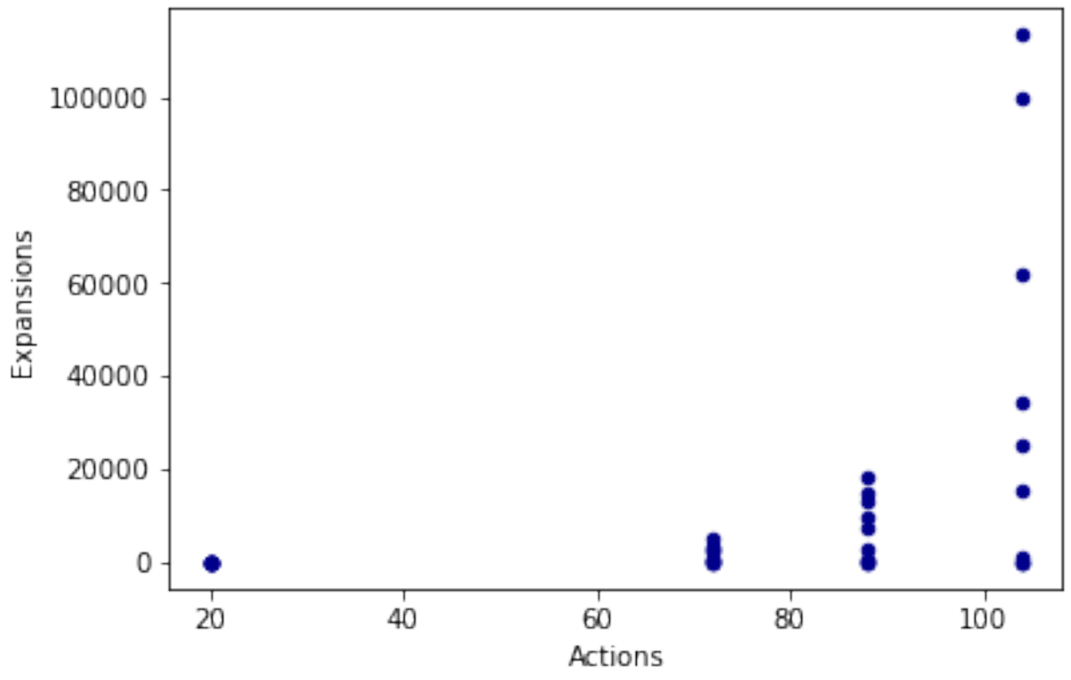
	Problem	Search	Actions	Expansions	Goal
0	AirCargoProblem1	breadth_first_search	20	43	
1	AirCargoProblem1	depth_first_graph_search	20	21	
2	AirCargoProblem1	uniform_cost_search	20	60	
3	AirCargoProblem1	greedy_best_first_graph_searchwithh_unmet_goals	20	7	
4	AirCargoProblem1	greedy_best_first_graph_searchwithh_pg_levelsum	20	6	

1.Use a table or chart to analyze the number of nodes expanded against number of actions in the domain

Answer: There is a positive correlation between the two factors.

In [14]:

```
df.plot.scatter(x="Actions", y="Expansions", c='DarkBlue')
plt.show()
```

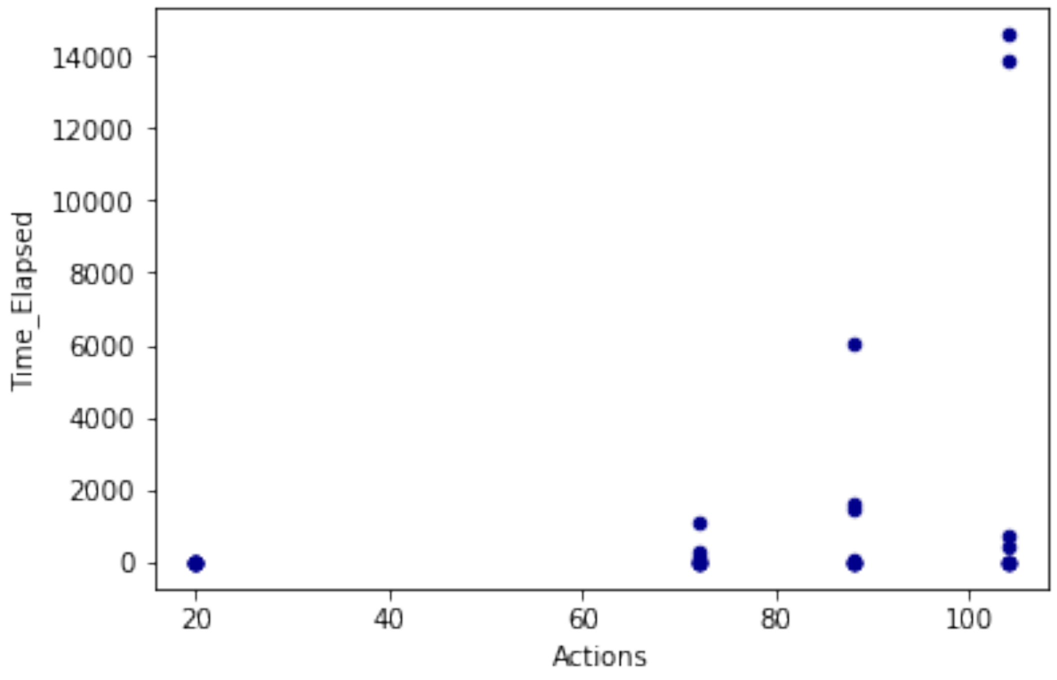


2.Use a table or chart to analyze the search time against the number of actions in the domain

There is a positive correlation between the two factors.

In [15]:

```
df.plot.scatter(x="Actions", y="Time_Elapsed", c='DarkBlue')
plt.show()
```



3.Use a table or chart to analyze the length of the plans returned by each algorithm on all search problems

Answer: The depth_first_graph_search generate longest plan length and then is the greedy_best_first_graph_search.

In [40]:

```
df.pivot(index="Search", columns="Problem", values="Plan_Length")
```

Out[40]:

Problem	**AirCargoProblem1**	**AirCargoProblem2**	**AirCa**
Search			
astar_searchwithh_pg_levelsum	6.0	9.0	
astar_searchwithh_pg_maxlevel	6.0	9.0	
astar_searchwithh_pg_setlevel	6.0	9.0	
astar_searchwithh_unmet_goals	6.0	9.0	
breadth_first_search	6.0	9.0	
depth_first_graph_search	20.0	619.0	
greedy_best_first_graph_searchwithh_pg_levelsum	6.0	9.0	
greedy_best_first_graph_searchwithh_pg_maxlevel	6.0	9.0	
greedy_best_first_graph_searchwithh_pg_setlevel	6.0	10.0	
greedy_best_first_graph_searchwithh_unmet_goals	6.0	9.0	
uniform_cost_search	6.0	9.0	

4.Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

Answer: greedy_best_first_graph_searchwithh_unmet_goals or depth_first_graph_search

In [52]:

```
df.groupby(["Problem"]).min()["Actions"]
```

Out[52]:

```
Problem
AirCargoProblem1     20
AirCargoProblem2     72
AirCargoProblem3     88
AirCargoProblem4    104
Name: Actions, dtype: int64
```

```
In [53]:
```

```
df.loc[df.Problem == "AirCargoProblem1",["Problem", "Search", "Time_Elapsed"]].sc
```

```
Out[53]:
```

	Problem	Search	Time_Elapsed
3	AirCargoProblem1	greedy_best_first_graph_searchwithh_unmet_goals	0.001875
1	AirCargoProblem1	depth_first_graph_search	0.006557
7	AirCargoProblem1	astar_searchwithh_unmet_goals	0.013105
2	AirCargoProblem1	uniform_cost_search	0.016912
0	AirCargoProblem1	breadth_first_search	0.020445
5	AirCargoProblem1	greedy_best_first_graph_searchwithh_pg_maxlevel	0.234453
9	AirCargoProblem1	astar_searchwithh_pg_maxlevel	0.322859
8	AirCargoProblem1	astar_searchwithh_pg_levelsum	0.360759
4	AirCargoProblem1	greedy_best_first_graph_searchwithh_pg_levelsum	0.666980
10	AirCargoProblem1	astar_searchwithh_pg_setlevel	1.031665
6	AirCargoProblem1	greedy_best_first_graph_searchwithh_pg_setlevel	1.148178

5.Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)

Answer: greedy_best_first_graph_searchwithh_unmet_goals.

In [54]:

```
df.loc[df.Problem == "AirCargoProblem4",["Problem", "Search", "Time_Elapsed"]].sc
```

Out[54]:

	Problem	Search	Time_Elapsed
36	AirCargoProblem4	greedy_best_first_graph_searchwithh_unmet_goals	0.024215
40	AirCargoProblem4	astar_searchwithh_unmet_goals	4.126687
33	AirCargoProblem4	breadth_first_search	5.505404
37	AirCargoProblem4	greedy_best_first_graph_searchwithh_pg_levelsum	6.883281
35	AirCargoProblem4	uniform_cost_search	8.394633
38	AirCargoProblem4	greedy_best_first_graph_searchwithh_pg_maxlevel	14.741153
41	AirCargoProblem4	astar_searchwithh_pg_levelsum	437.795106
34	AirCargoProblem4	depth_first_graph_search	767.090839
39	AirCargoProblem4	greedy_best_first_graph_searchwithh_pg_setlevel	13833.821295
42	AirCargoProblem4	astar_searchwithh_pg_maxlevel	14585.299921

6.Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Answer: astar_searchwithh_unmet_goals, breadth_first_search, uniform_cost_search

In [ ]:

```
```