Lab 1: Implementing BFS and DFS

In [1]:

```
from time import time
import sys
import matplotlib.pyplot as plt
```

In [2]:

```
graphOne = {
  0 : [1, 2],
  1 : [2, 3],
  2 : [3],
  3 : []
}
```

In [3]:

```
visited = []
queue = []

def bfs(visited, graph, node):
  visited.append(node)
  queue.append(node)

  while queue:
    s = queue.pop(0)
    print(s)

    for neighbour in graph[s]:
      if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)

visitedNodes = []

def DFS(visitedNodes, graph, currentNode):
    if currentNode not in visitedNodes:
        print(currentNode)
        visitedNodes.append(currentNode)

        for neighbour in graph[currentNode]:
            DFS(visitedNodes, graph, neighbour)
```

In [4]:

```
start = int(input("\nStarting node for BFS: "))

t0 = time()
bfs(visited, graphOne, start)
t1 = time()

dfstime = t1-t0
print("BFS timed at {:3f} seconds".format(t1-t0))
print("BFS took {} bytes".format(sys.getsizeof(visited) + sys.getsizeof(queue)))
```

```
Starting node for BFS: 1
1
2
3
BFS timed at 0.000789 seconds
BFS took 144 bytes
```

In [5]:

```
start = int(input("\nStarting node for DFS: "))

t0 = time()
DFS(visitedNodes, graphOne, start)
t1 = time()

bfstime = t1-t0
print("DFS timed at {:.3f} seconds".format(t1-t0))
print("DFS took {} bytes".format(sys.getsizeof(visitedNodes)))
```

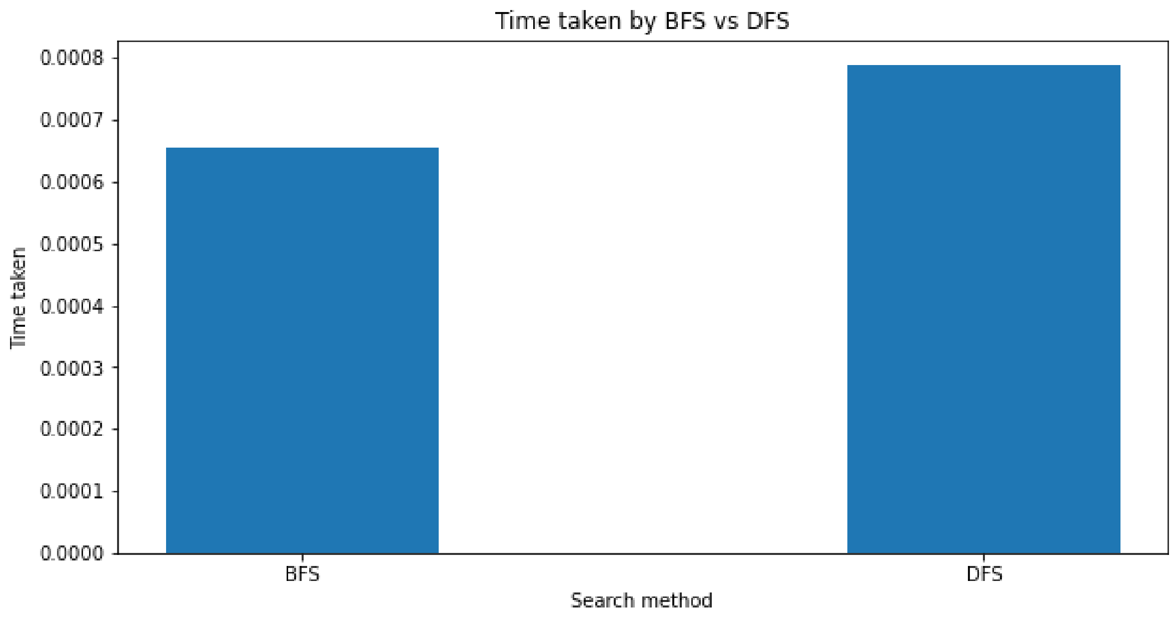
```
Starting node for DFS: 1
1
2
3
DFS timed at 0.001 seconds
DFS took 88 bytes
```

In [6]:

```
data = {'BFS': bfstime, 'DFS': dfstime}
method = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

plt.bar(method, values, width = 0.4)
plt.xlabel("Search method")
plt.ylabel("Time taken")
plt.title("Time taken by BFS vs DFS")
plt.show()
```

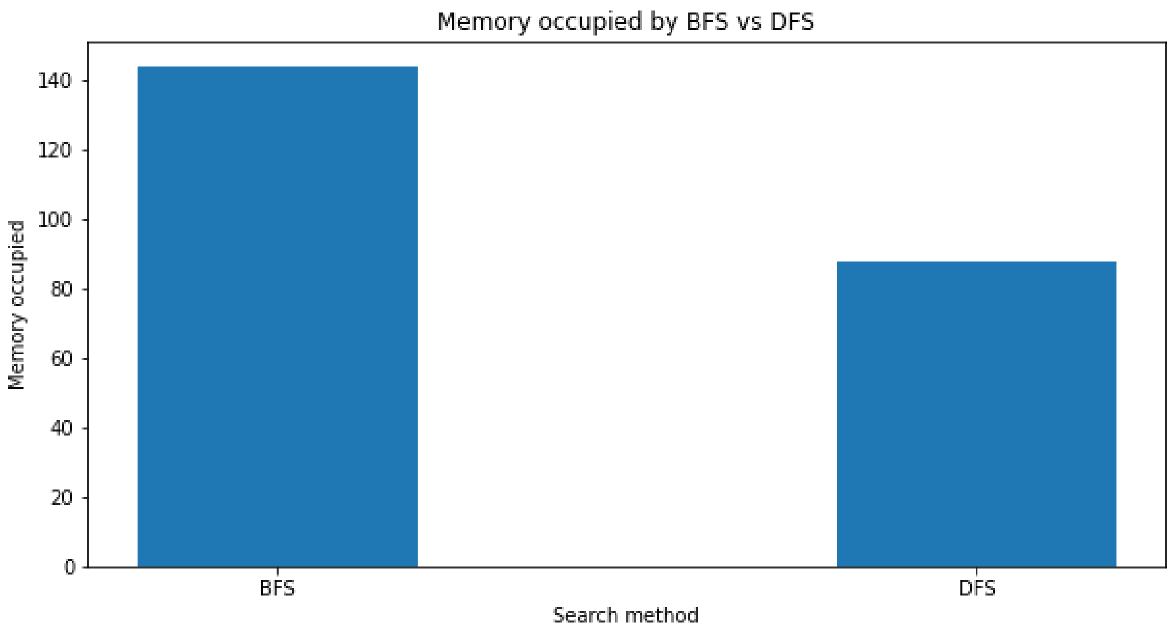


In [7]:

```
data = {'BFS': (sys.getsizeof(visited) + sys.getsizeof(queue)), 'DFS': (sys.getsize
method = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

plt.bar(method, values, width = 0.4)
plt.xlabel("Search method")
plt.ylabel("Memory occupied")
plt.title("Memory occupied by BFS vs DFS")
plt.show()
```


Memory occupied by BFS vs DFS
Memory occupied
BFS
DFS
Search method