124 . Approximation Algorithm Vertex Cover, Set Cover

AIM: To soleve the Vertex Cover, Set Cover by using Approximation Algorithm

PROGRAM:

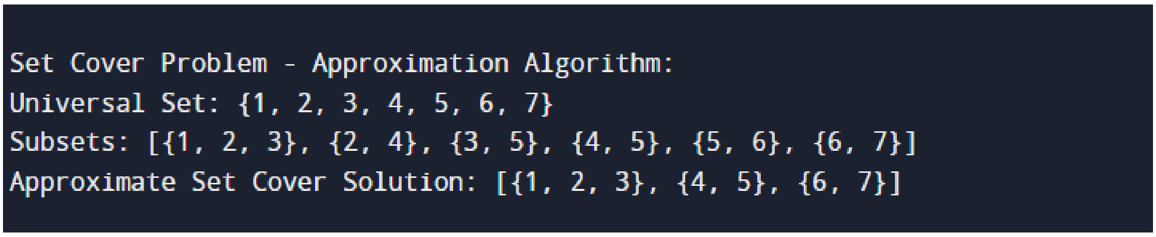
```
def set_cover_approx(universal_set, subsets):
    """ Approximation algorithm for Set Cover Problem using a greedy approach """
    elements_uncovered = set(universal_set)
    solution = []

    while elements_uncovered:
        best_subset = None
        max_covered = set()

        for subset in subsets:
            covered = elements_uncovered.intersection(subset)
            if len(covered) > len(max_covered):
                max_covered = covered
                best_subset = subset
        if not best_subset:
            break
        solution.append(best_subset)
        elements_uncovered -= max_covered
    return solution
universal_set = {1, 2, 3, 4, 5, 6, 7}
subsets = [{1, 2, 3}, {2, 4}, {3, 5}, {4, 5}, {5, 6}, {6, 7}]

print("\nSet Cover Problem - Approximation Algorithm:")
print("Universal Set:", universal_set)
print("Subsets:", subsets)
set_cover_solution = set_cover_approx(universal_set, subsets)
print("Approximate Set Cover Solution:", set_cover_solution)
```

OUTPUT:



```
Set Cover Problem - Approximation Algorithm:
Universal Set: {1, 2, 3, 4, 5, 6, 7}
Subsets: [{1, 2, 3}, {2, 4}, {3, 5}, {4, 5}, {5, 6}, {6, 7}]
Approximate Set Cover Solution: [{1, 2, 3}, {4, 5}, {6, 7}]
```

TIME COMPLEXITY: O (n.m)