let the amount be t.
this is the change-making problem https://en.wikipedia.org/wiki/Change-making_problem, which is NPC.
1. dfs.
2. knapsack, DP. $O(nt)$.
3. set a threshold b. for coins with values $> b$, we will use at most $\frac{t}{b}$ (with multiplicity) coins, use FFT and recursion to compute subset sum with cardinality information, in $O(t \cdot \frac{t}{b} \log^2 t)$. for coins with value $\leq b$, there are at most b distinct coin values, perform DP for knapsack in $O(bt)$. set $b = \sqrt{t} \log t$, the total time is $O(\frac{t}{b} \cdot t \log^2 t + bt) = O(t\sqrt{t} \log t)$.
4. FFT+doubling, each FFT takes $O(t \log t)$, need to perform $O(\log t)$ doubling steps. $O(t \log^2 t)$.
5. deterministic $O(t \log t \log \log t)$ and randomized $O(t \log t)$. see my paper [1].
6. $\tilde{O}(u)$, where u is the maximum coin value. see my paper (to be published).

remark. the all-values change-making problem can also be solved efficiently, in $\tilde{O}(t^{4/3})$ and $O(u^2 \log u + t)$ time. see my paper (to be published).

References

[1] Timothy M. Chan and Qizheng He. On the change-making problem. In *Symposium on Simplicity in Algorithms*, pages 38–42. SIAM, 2020.