

meet in the middle+hashing: for $i = 1, \dots, n$, store sum of all pairs in $a[1 \dots i - 1]$ in the hash table, query sum of $-a[i] - a[j]$ for $j = i + 1, \dots, n$. $O(n^2)$ expected, or $O(n^2 \log n)$ deterministic by sorting.
for k -sum, the best known algorithm is $O(n^{\lceil d/2 \rceil} / \text{poly} \log n)$. can reduce to 2-sum when k is even, or reduce to an unbalanced 3-sum when k is odd.

<https://cs.stackexchange.com/questions/2973/generalised-3sum-k-sum-problem>

lower bound: no $n^{o(k)}$ algorithm for k -sum, if Strong ETH holds [1].

References

- [1] Mihai Pătraşcu and Ryan Williams. On the possibility of faster sat algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1065–1075. SIAM, 2010.