ابتدا خانه های جدول را به صورت زیر

شماره گذاری می کنیم و مقدار می دهیم (یک به یک و پوشا)

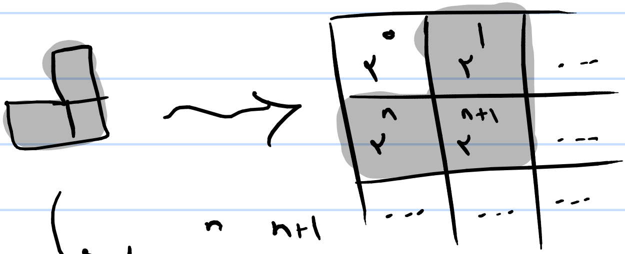
2^0	...	2^n
2^{n+1}	...	2^n
...	...	...
2^{n^2-n}	...	2^{n^2-1}

حال با گذاشتن هر شکل در بالاترین و چپ ترین

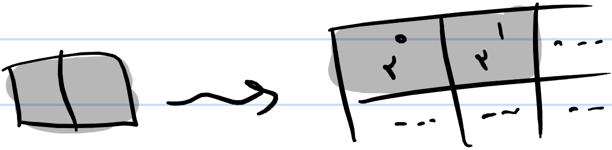
حالت ممکن در جدول مقادیر ۴ خانه های ۱،

که پر می کند با هم جمع می کنیم و به آن شکل نسبت

می‌دهیم. (می‌دانیم که نمایش باینری یک
عدد می‌باشه، چون اشکال به صورت دهنده‌ها باهم
متفاوت اند این ثبت دارن که به یک ورودیش
است.) برای مثال:



$2 + 2^n + 2^{n+1}$



$1 + 2 = 3$

نسبت دادن و پیدا کردن اعداد منسوب به شکل

چند جمله‌ای خواهد بود. زیرا به تعداد هر مربع 1×1

در شکل باید عدد مربوطه را پیدا کرده و جمع کنیم.

$\left[O(n^4) \leftarrow O(\text{تعداد شکل‌ها} \times \text{تعداد مربع‌های سوخته برای هر شکل}) \right]$

حال با ضرب هر عدد k (که k عضو

توان‌های 2^i به شرطی که $0 \le i < n^2$ باشد.)

می‌توان شکل را طوری منتقل کرد که

مربع بالا چپ آن در خانه خطیر k قرار بگیرد.

(در واقع مربع مجازی که از بالا و چپ مماس بر شکل باشد.)

حال به ازای هر k ممکن که شکل جاری
را نتوان منتقل کرد و شکل از جدول
بیرون نزند، k را در عدد شکل نظیر ضرب
کرده و آن را ورودی مسئله subset
می‌کنیم. (k بیش از n^2 نخواهد شد)
بنابراین تولید تمامی حالات ممکن $O(n^6)$
خواهد بود.

هزینه مسئله subset sum: $\sum_{i \in [0, n^2]} 2^i = 2^{n^2+1} - 1$

خروجی مسئله Subset Sum یک مجموعه
عدد است که با توجه به یک به یک بودن نظیرسازی
اعداد به اشکال، می‌توان اشکال و جای
آن‌ها را (با داشتن ضریب k) حساب
کرد و آن را به صورت خروجی گزارش نمود.
بنابراین هر ۲ تابع معرفی شده چندجمله‌ای
بود و مسئله حل می‌شود.