a) $\left\{ {n \atop k} \right\}$ to ilość podziałów zbioru n-elementowego na k niepustych podzbiorów.
Zastanówmy się jak można to policzyć?

Załóżmy, że mamy n ponumerowanych piłeczek, oraz k pudełek do których chemy je upchnąć w taki sposób, żeby żadno z pudełek nie zostało puste.

Ustawmy sobie piłeczki w kolejności według numerów. Teraz, spośród nich, wybieramy sobie pierwszą, oraz dowolne $k-1$ innych. Będą one gwarantami tego, że w każdym pudełku coś się znajdzie. Pierwsza wybrana pójdzie do pierwszego pudełka, druga do drugiego itd. Piłeczki pomiędzy i-tą a $(i+1)$szą wybraną piłeczką w losowy sposób rozłożymy do pierwszych i pudełek (jeśli zostały nam jakieś na koniec, to rozlosowujemy je po wszystkich). No dobrze, w ten sposób na pewno uzyskamy jakiś podział zbioru na bloki. Pytanie czy każdy i czy coś nam się przypadkiem nie powtórzy.

Weźmy zatem jakiś dowolny podział naszego zbioru. Składa się on z k podzbiorów. Ponieważ n jest skończone, to każdy z nich posiada element najmniejszy, w szczególności - w którymś z nich jest to piłeczka z numerem 1. Elementy te stanowią jeden z układów jaki mogliśmy wybrać w naszym algorytmi podziału, więc póki co jest dobrze. Co z resztą? W każdym podzbiorze oprócz reprezentanta znajdują się tylko piłeczki z numerem większym od niego. Analogicznie jest w naszym algorytmie - jeśli wybraliśmy jakiś zbiór wyszczególnionych piłeczek, to w pudełku z którąś z nich mogą się pojawić dowolne o numerze większym. Więc istnieje w naszym algorytmie sytuacja, w której uzyskujemy rządany podział.

Jeśli zaś chodzi o powtórzenia - to jasne jest, że nie mogą one wystąpić po etapie wyboru reprezentantów pudełek. Ale wcześniej jest to także niemożliwe, gdyż wybór różnych reprezentantów od razu uniemożliwia uzyskanie tego samego układu.

Zatem szczęśliwie algorytm jest równoważny ze znalezieniem wszystkich podziałów i ilość układów jakie generuje odpowiada szukanej liczbie Stirlinga. Czas je policzyć.

Zauważmy, że grono reprezentantów jest jednoznacznie wyznaczone przez przedziały pomiędzy nimi. Nazwijmy je $a_1, a_2 ... a_k$ (a_k jest przedziałem między ostatnim reprezentantem, a końcem naszego posortowanego szeregu piłeczek). Wiadomo, że $a_1 + a_2 + ... + a_k = n - k$ (gdzie $n-k$ nas wyraźnie cieszy nawiązując do szukanego wyniku) oraz $a_i \geq 0$. Każdy przedział a_i możemy rozdysponować między pudełka na i^{a_i} sposobów. Zatem sumując po wszystkich możliwych przedziałach uzyskujemy:

$$\sum_{\substack{a_1+a_2+...+a_k=n-k \\ a_1,...,a_k \geq 0}} 1^{a_1} \cdot 2^{a_2} \cdot ... \cdot k^{a_k}$$

Teraz, jak rozwiniemy nasze iloczyny potęg w jeden długi iloczyn okaże się, że są one dość specyficzne. Otóż $1^{a_1} \cdot 2^{a_2} \cdot ... \cdot k^{a_k}$ jest iloczynem $n-k$ składników, z których każdy jest większy bądź równy poprzedniemu. A ogólniej - otrzymamy pod sumą, wymnożone wyrazami, wszystkie możliwe ciągi rosnące długości $n-k$

o wyrazach ze zbioru $1, ...k$. W naturalny więc sposób przechodzimy do postaci ostatecznej:

$$\sum_{1 \leq i_1 \leq ... \leq i_{n-k} \leq k} i_1 \cdot i_2 \cdot ... \cdot i_k$$

b) $\begin{bmatrix} n \\ k \end{bmatrix}$ to ilość podziałów zbioru n-elementowego na k cykli.

Przypadek różni się od poprzedniego tym, że w obrębie podzbioru, możemy przemieszać elementy poza najmniejszym dostając różne (tzn. nie równoważne sobie) cykle.

Spróbujmy zatem w jakiś sposób poumieszczać n piłeczek w k pudełkach, z których każde ma w sobie przegródki powodujące, że w obrębie pudełka istnieje kolejność włożonych w niego elementów. Najfaniej by było, gdybyśmy dostali postać normalną, tzn. w jednoznaczny sposób reprezentującą cykl. Osiągniemy to np. ustalając, że najmniejszy element w cyklu stoi na pierwszym miejscu. Teraz ponumerujmy sobie nasze piłeczki i ustawmy kolejno w rzędzie.
Piłeczkę z numerem 1 musimy wsadzić do któregoś pudełka, do pierwszej przegródki, bo jeśli jest w jakimś cyklu, to na pewno będzie elementem najmniejszym. Wsadźmy więc do pierwszego wolnego. To był nasz krok zerowy. Zastanówmy się co możemy teraz zrobić. Na pewno jak wsadzimy jakąkolwiek piłeczkę do naszego pudełka, to nic się nie popsuje. Problem pojawia się jak zechcemy wsadzić coś do nowego - jeśli umieścimy cokolwiek w pierwszej przegródce, to w kolejne możemy wstawiać tylko piłeczki o numerach większych. To powodowałoby spore zamieszanie. Tak samo jak analizowanie co się dzieje w paru pudełkach na raz. Dlatego uprościmy sobie pracę i określimy sobie następujące zasady wkładania elementów:

- na raz zajmujemy się tylko jednym pudełkiem
- do pudełka wkładamy elementy w dowolnej kolejności, aż uznamy, że starczy
- jeśli uznaliśmy, że starczy, to odkładamy pudełko na bok i uznajemy za wypełnione. Bierzemy wtedy kolejne puste pudełko i umieszczamy w nim najmniejszy element z pozostałej puli piłeczek (kierując się taką samą logiką, jak przy piłeczce 1)

Oczywiście, od razu pojawiają się pewne problemy. Na przykład co jak na zbraknie piłeczek? Dlatego należy dodać:

- jeśli zostało tyle piłeczek co pustych pudełek, to do każdego wkładamy po jednej z nich i kończymy rozdzielanie

W ten sposób mamy gwarancję, że żadne pudełko nie będzie puste. I na pewno uzyskamy *jakiś* podział na cykle w postaci normalnej (bo zawsze na pierwszym miejscu w pudełku jest piłka z lokalnie najmniejszym numerem).

Zastanówmy sie teraz, czy stosując tą metodę na wszystkie możliwe sposoby wypełnimy przestrzeń rozkładów na cykle. weźmy dowolny taki rozkład. Trywialnie widać, że można go uzyskać idąc od lewej do prawej i wybierając do pudełek

liczby jakie się w nim pojawiają oraz zamykając pudełka gdy natrafimy na koniec cyklu. Ok... ale może dokonując różnych wyborów możemy uzyskać ten sam układ? Widać jednak, że algorytm „zabezpiecza tyły" dzięki postaci normalnej i dwa układy są takie same, tylko jeśli postawimy jeden pod drugim i w każdym miejscu jest tak samo. I gdy dokładamy jakiś element, to wybór dwóch różnych od razu powoduje różnicę układów wygenerowanych przez te dwie ścieżki, dokładnie na tej pozycji. Gdy zdecydujemy się w jednej ścieżce zamknąć pudełko a w drugiej dołożyć element, to też jest inaczej, bo różnica na pewno się pojawia między cyklem, który właśnie zamknęliśmy a jego odpowiednikiem w drugiej ścieżce. Wiecej ruchów nie ma. Więc za każdym razem generujemy inny układ. I to nas bardzo cieszy, bo właśnie wyjaśniliśmy, że ilość układów generowana przez algorytm jest równa szukanej liczbie.

To teraz spróbujmy połączyć ten opis metody z jakimiś wzorkami. Najlepiej jakimiś, które opisują generowane przez nas układy. Łatwo zauważyć, że wśród wszystkich podziałów jakie będziemy w ten sposób generować można wyodrębnić klasy, ze względu na numery kroków w których zamykamy pudełka. Jest to jakieś $k-1$ (bo wspólną wszystkim układom piłeczkę 1 możemy pominąć) liczb z zakresu $\{2, ..., n\}$. To ile jest układów w danej klasie? Tyle co możliwości wybierania elementów w krokach pomiędzy zamknięciami pudełek. W i-tym kroku zawsze dysponujemy w puli $n-i$ piłeczkami. I na tyleż sposobów możemy z nich wybrać element. Zatem należy wziąć zbiór P zawierający numery kroków w których dokonujemy wyboru piłeczek do wsadzenia (jest to $\{2, .., n\}$ z wyjętymi $n-1$ krokami zamknięcia pudełek) i policzyć iloczyn $\prod_{i \in P}(n-i)$. Jest on iloczynem $n-k$ różnych liczb całkowitych z zakresu $\{1, ..., n-1\}$. Jest on różny dla każdej klasy układów i razem wyczerpują one wszystkie postacie jakie może przybrać. Teraz wystarczy przesumować wszystkie klasy i dostajemy co następuje:

$$\sum_{0<i_1<...<i_{n-k}<n} i_1 \cdot ... \cdot i_{n-k}$$