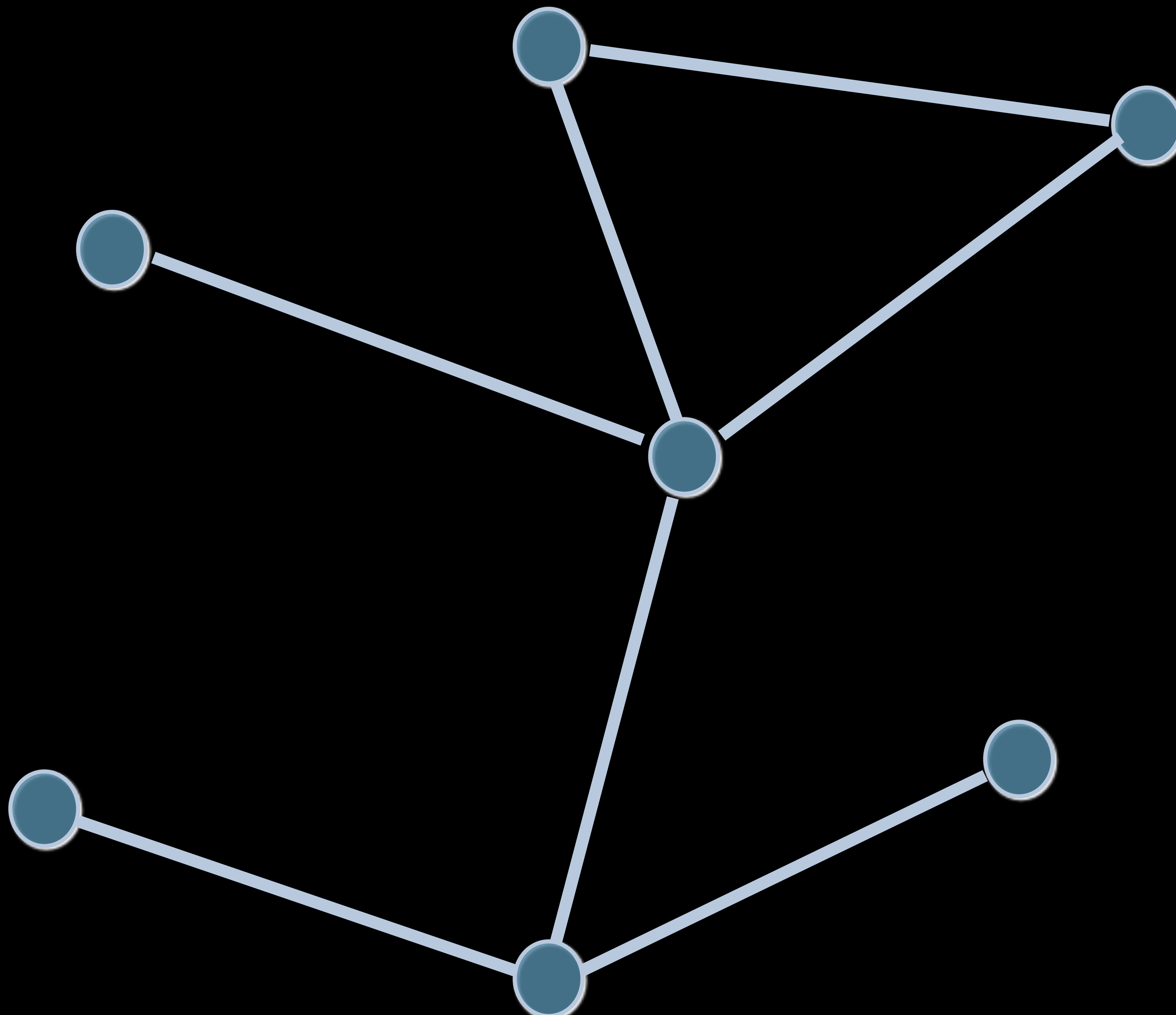
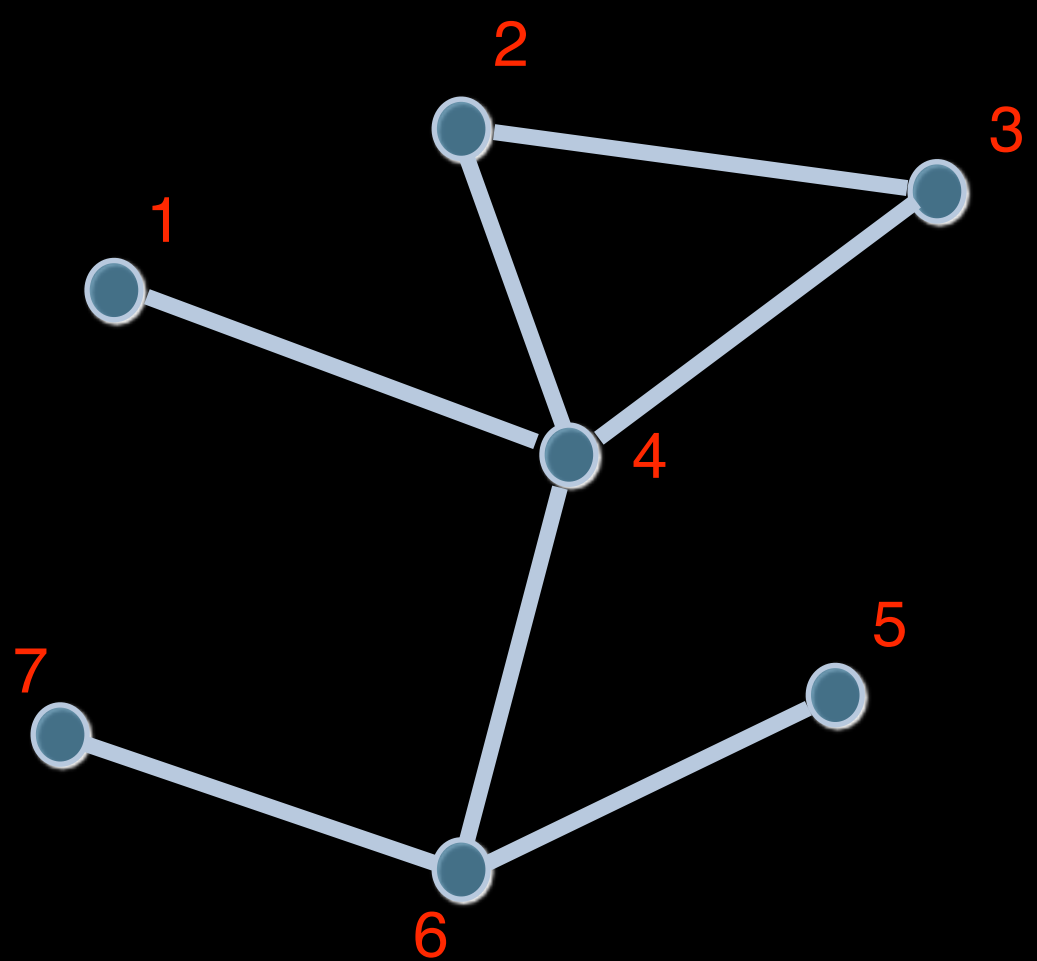
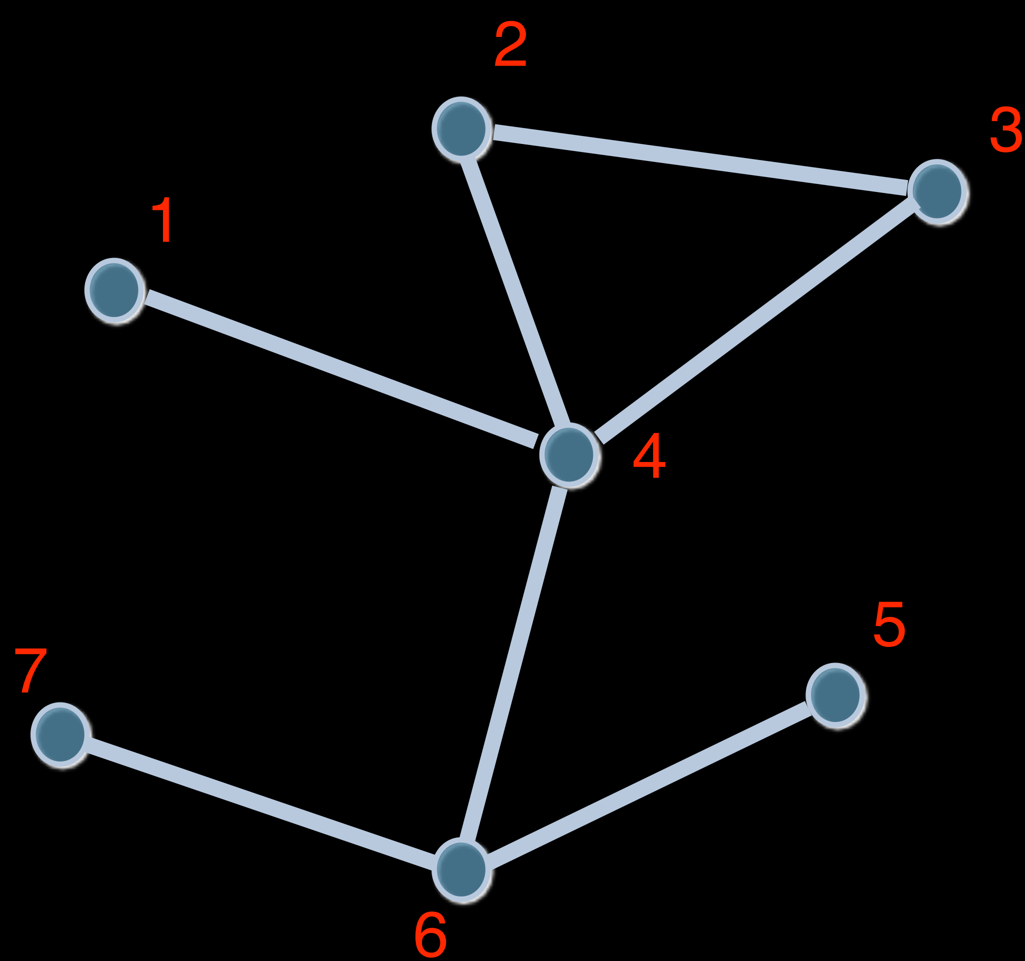
The skew spectrum of graphs

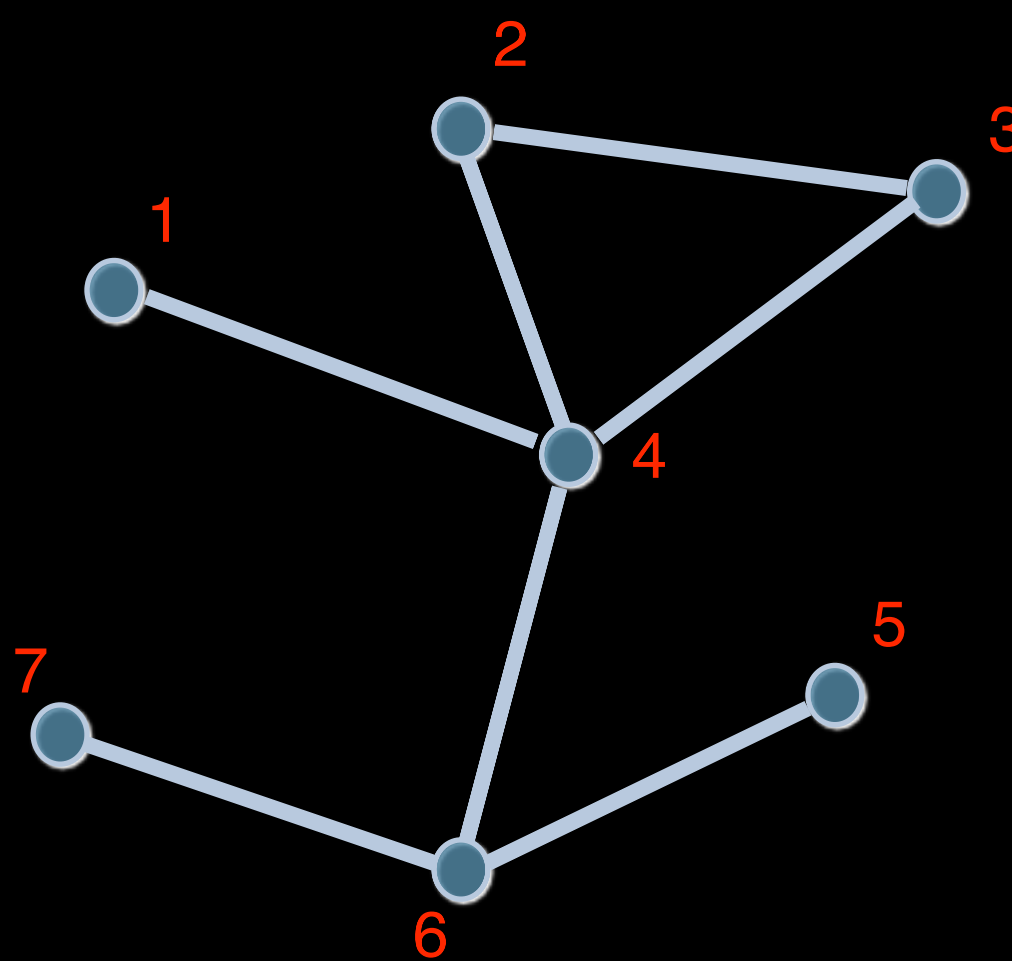
Risi Kondor
Gatsby Unit, UCL
with
Karsten Borgwardt
University of Cambridge




2
3
1
4
5
7
6



$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$



$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$q(A)$ is a graph invariant if it is invariant to relabeling.

$\mathrm{poly}(n)$ time computable complete set of invariants

Graph isomorphism problem

efficiently computable set of invariant features

Graph kernels, etc.

$$f(\sigma) = [A]_{\sigma(n),\sigma(n-1)}$$

$$f : \mathbb{S}_n \to \mathbb{R}$$

$$f(7\ 6\ ?\ ?\ ?\ ?\ ?) = [A]_{1,2}$$
$$f(7\ ?\ 6\ ?\ ?\ ?\ ?) = [A]_{1,3}$$
$$f(7\ ?\ ?\ 6\ ?\ ?\ ?) = [A]_{1,4}$$
$$\vdots \qquad\qquad \vdots$$
$$f(?\ 7\ 6\ ?\ ?\ ?\ ?) = [A]_{2,3}$$
$$f(?\ 7\ ?\ 6\ ?\ ?\ ?) = [A]_{2,4}$$
$$\vdots \qquad\qquad \vdots$$

Now if we permute the vertices by $i \mapsto \pi(i)$ ....

$$[A']_{\pi(i),\pi(j)} = [A]_{i,j}$$

Now if we permute the vertices by $i \mapsto \pi(i)$ ....

$$[A']_{\pi(i),\pi(j)} = [A]_{i,j}$$

$$f'(?\ 7\ ?\ 6\ ?\ ?\ ?) =$$

$\uparrow$ $\pi(j)$ $\quad$ $\uparrow$ $\pi(i)$

$$f(?\ 7\ ?\ 6\ ?\ ?\ ?)$$

$\uparrow$ i $\quad$ $\uparrow$ j

... in other words $f'(\pi\sigma) = f(\sigma)$.

... or $f' = f^\pi$, where

$$f^\pi(\sigma) = f(\pi^{-1}\sigma)$$

is the translate of f by π.

2. Non-commutative harmonic analysis and invariants

G is a group if for any $x, y, z \in G$

1. $xy \in G$,
2. $x(yz) = (xy)z$,
3. there is an $e \in G$ such that $ex = xe = x$,
4. there is an $x^{-1} \in G$ such that $xx^{-1} = x^{-1}x = e$.

Permutations $\sigma\colon \{1, 2, \ldots, n\} \to \{ 1, 2, \ldots, n\}$ form a group called the symmetric group, denoted $\mathbb{S}_n$.

$$\widehat{f}(k) = \sum_{x=0}^{n-1} e^{-ikx} f(x)$$

$$\rho(x)\,\rho(y) = \rho(xy)$$

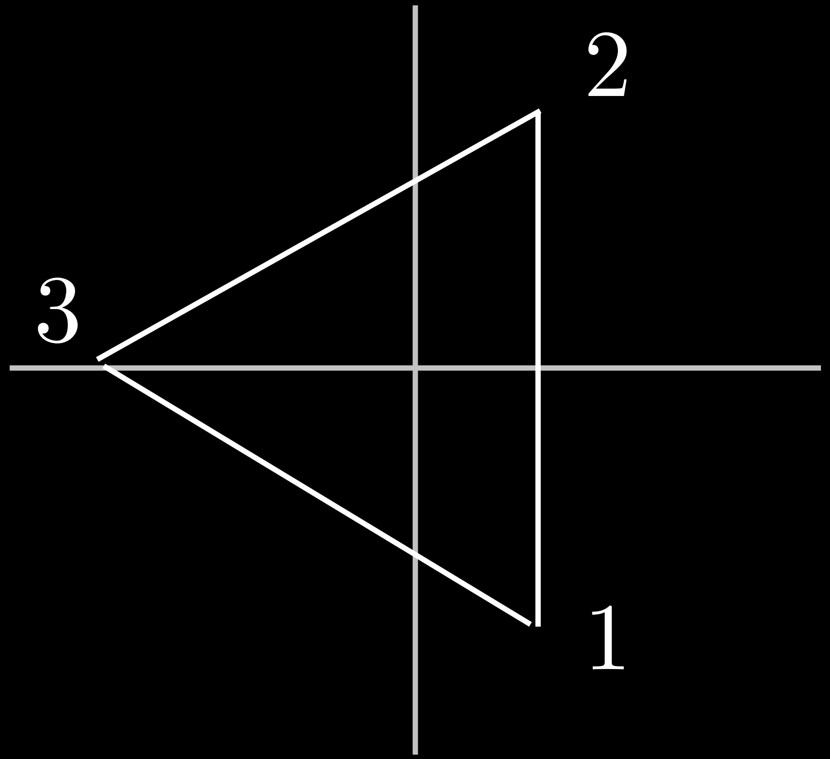
$$\widehat{f}(\rho) = \sum_{x \in G} \rho(x)\, f(x)$$

$$\rho(x)\,\rho(y) = \rho(xy)$$

$\rho\colon G \to \mathbb{C}^{d\times d}$ is called a representation of G

Example

$$\mathbb{S}_3$$



$$\rho(e) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\rho((12)) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\rho((123)) = \begin{pmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{pmatrix}$$

Equivalence:

$$\rho_1(x) = T^{-1} \rho_2(x)\, T$$

Reducibility:

$$\rho(x)\, \rho(y) = \rho(xy)$$

$$T^{-1} \rho(x)\, T = \begin{pmatrix} \rho_1(x) & 0 \\ 0 & \rho_2(x) \end{pmatrix}$$

$\rho \colon G \to \mathbb{C}^{d \times d}$ is called a representation of G

A complete set of inequivalent irreducible unitary representations we denote $\mathcal{R}$.

The Fourier transform on a group is

$$\widehat{f}(\rho) = \sum_{x \in G} f(x)\,\rho(x) \qquad \rho \in \mathcal{R}$$

- Diaconis: Group representations in probability and statistics (1988)
- Clausen, Maslen, Rockmore, Healy, ... : FFTs
- Kondor, Howard and Jebara: Multi-object tracking with representations of the symmetric group (AISTATS, 2007)
- Huang, Guestrin and Guibas: Efficient inference for distributions on permutations (NIPS, 2007)

$$\widehat{f^t}(\rho) = \rho(t)\,\widehat{f}(\rho)$$

The power spectrum of f is the set of invariant matrices

$$\widehat{a}(\rho) = \widehat{f}(\rho)^\dagger \cdot \widehat{f}(\rho)$$

$$\widehat{a}^t(\rho) = (\rho(t)\widehat{f}(\rho))^\dagger \cdot (\rho(t)\widehat{f}(\rho)) = \widehat{f}(\rho)^\dagger \cdot \widehat{f}(\rho) = \widehat{a}(\rho)$$

Kakarala's non-commutative bispectrum is

$$b(\rho_1, \rho_2) = C^\dagger \left(\widehat{f}(\rho_1) \otimes \widehat{f}(\rho_2)\right)^\dagger C \bigoplus_\rho \widehat{f}(\rho)$$

where

$$\rho_1(z) \otimes \rho_2(z) = C \left[\bigoplus_\rho \rho(z)\right] C^\dagger$$

is the Clebsch-Gordan decomposition.

[Kakarala, 1992]

The skew spectrum is the unitarily equivalent, but easier to compute set of matrices

$$\widehat{q}_z(\rho) = \widehat{r}_z(\rho)^\dagger \cdot \widehat{f}(\rho)$$

where

$$r_z(x) = f(xz)f(x)$$

[Kondor, 2007]

3. Back to graphs...

What we have so far:

1. $f(\sigma) = [A]_{\sigma(n),\sigma(n-1)}$

2. Under permuting the vertices $f' = f^{\pi}$

3. Our favorite invariant is the skew spectrum

$$\widehat{q}_{\nu}(\rho) = \widehat{r}_{\nu}(\rho)^{\dagger} \cdot \widehat{f}(\rho) \qquad r_{\nu}(\sigma) = f(\sigma\nu)\, f(\sigma)$$

where

$$\widehat{f}(\rho) = \sum_{\sigma \in \mathbb{S}_n} \rho(\sigma)\, f(\sigma)$$

Far too expensive in this form!

$$f(7\ 6\ ?\ ?\ ?\ ?\ ?) = [A]_{1,2}$$

$$f(7\ ?\ 6\ ?\ ?\ ?\ ?) = [A]_{1,3}$$

$$f(7\ ?\ ?\ 6\ ?\ ?\ ?) = [A]_{1,4}$$

$$\vdots \qquad\qquad \vdots$$

$$f(?\ 7\ 6\ ?\ ?\ ?\ ?) = [A]_{2,3}$$

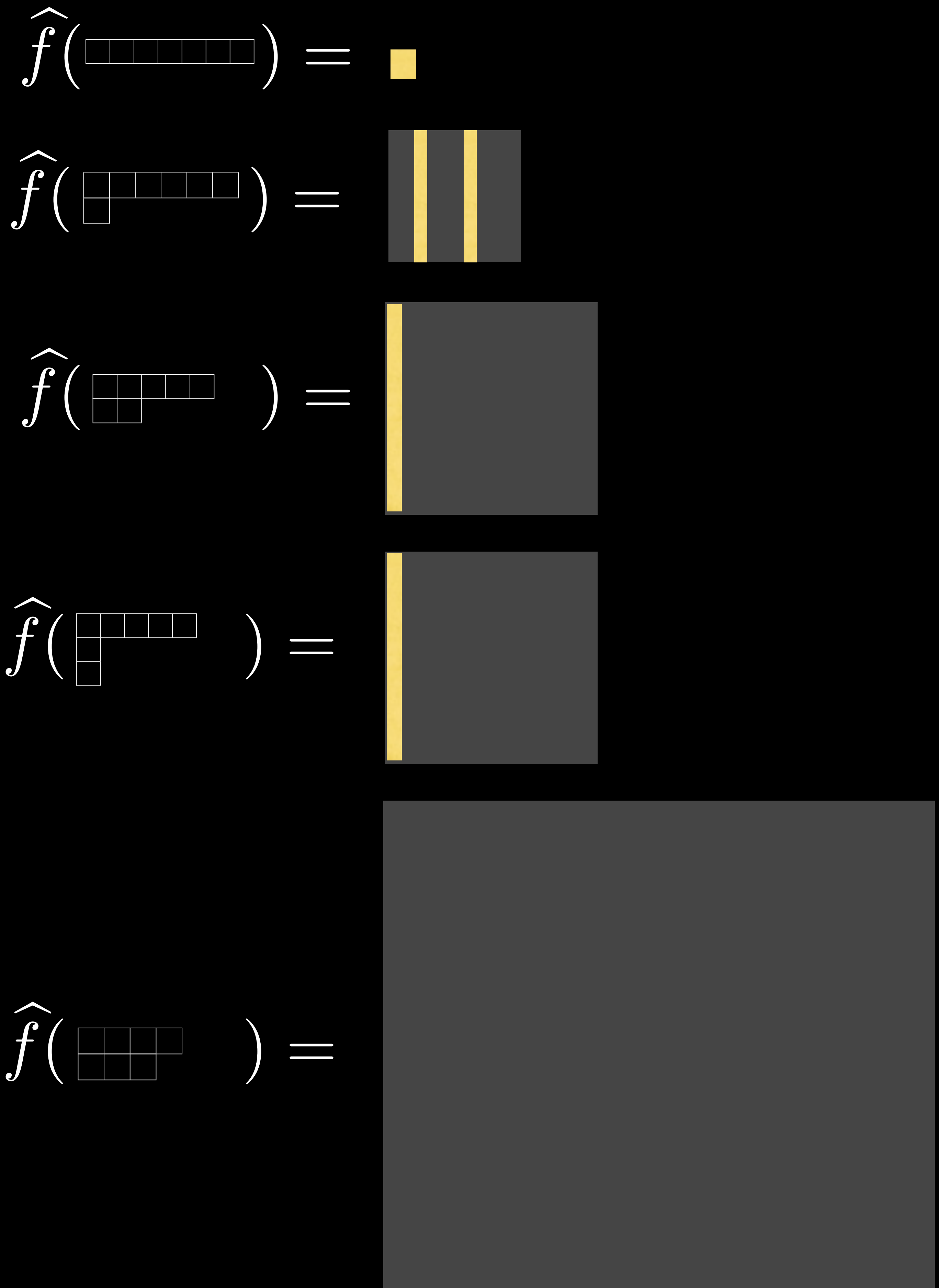
$$f(?\ 7\ ?\ 6\ ?\ ?\ ?) = [A]_{2,4}$$

$$\vdots \qquad\qquad \vdots$$

1. The ν index only has to extend over one representative from each $\mathbb{S}_{n-2}\,\sigma\,\mathbb{S}_{n-2}$ coset.

2. The $\widehat{f}$ and $\widehat{r}_\nu$ Fourier transforms are very sparse.

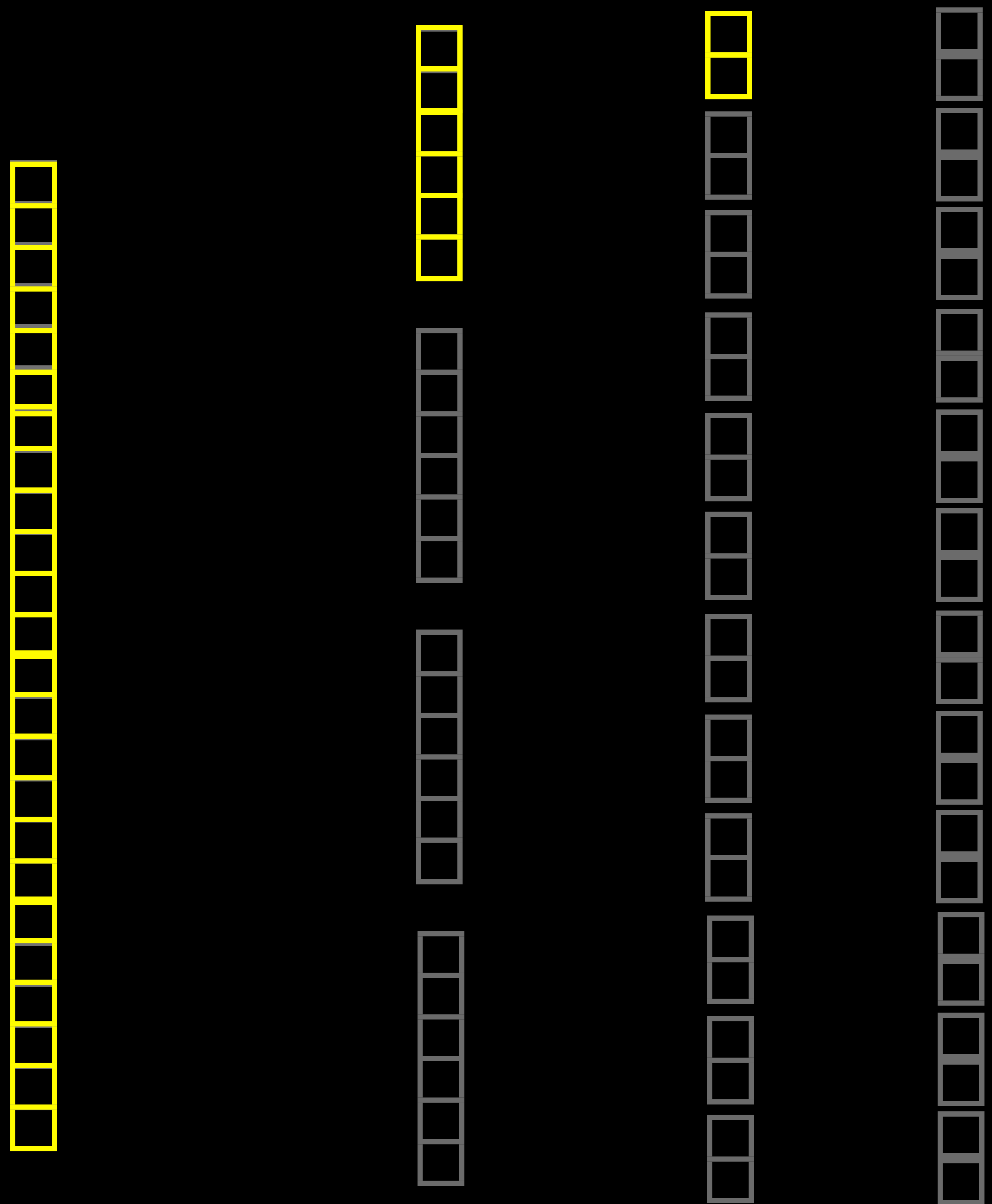
	$\widehat{r}_\nu(\rho)^\dagger \cdot \widehat{f}(\rho)$
$d = 1$	$1 \cdot 1$
$d = n - 1$	$2 \cdot 2$
$d = n(n-3)/2$	$1 \cdot 1$
$d = (n-1)(n-2)/2$	$1 \cdot 1$
	7
$d = n(n-1)(n-5)/6$	

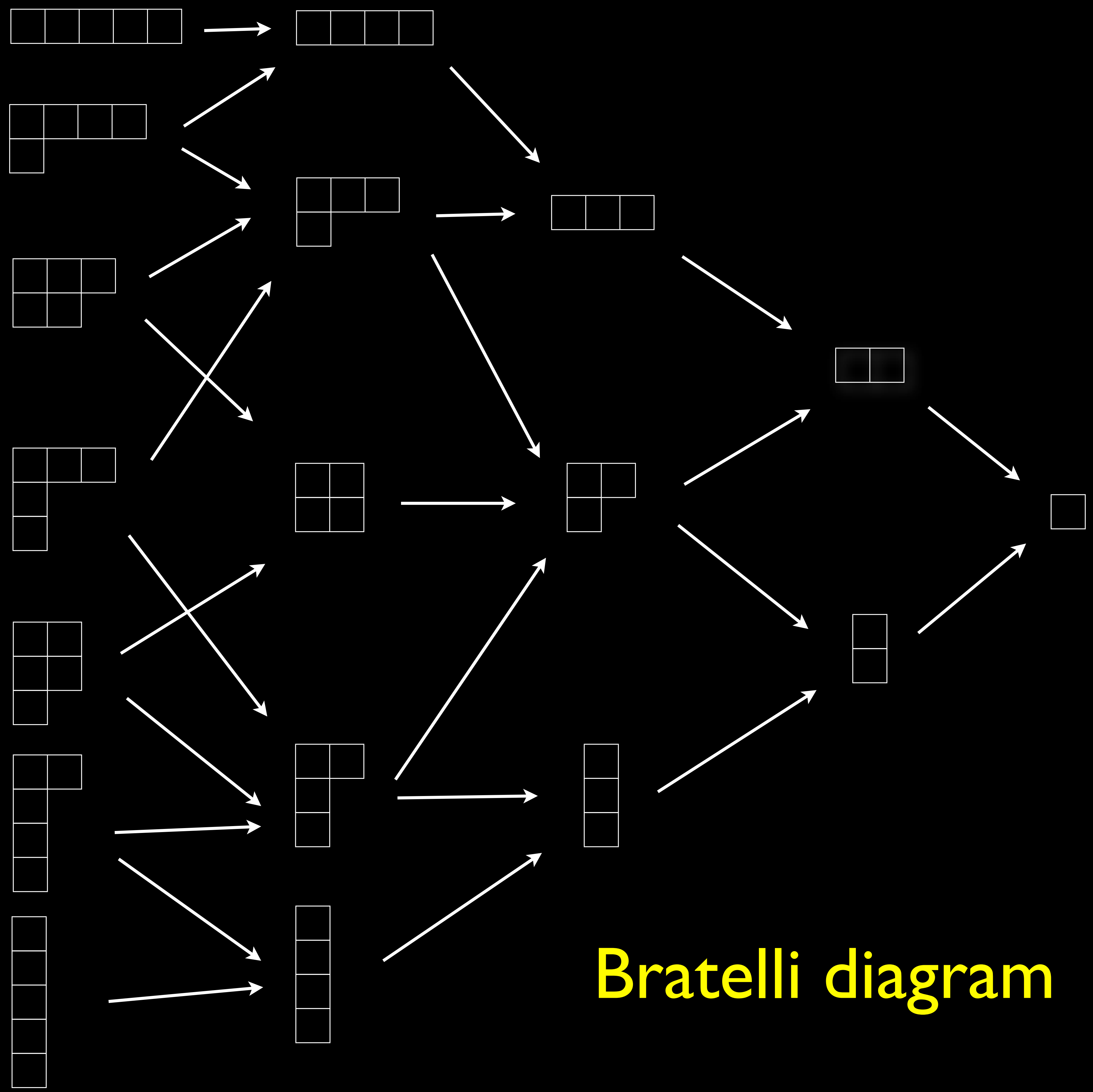


The answer is **49.**

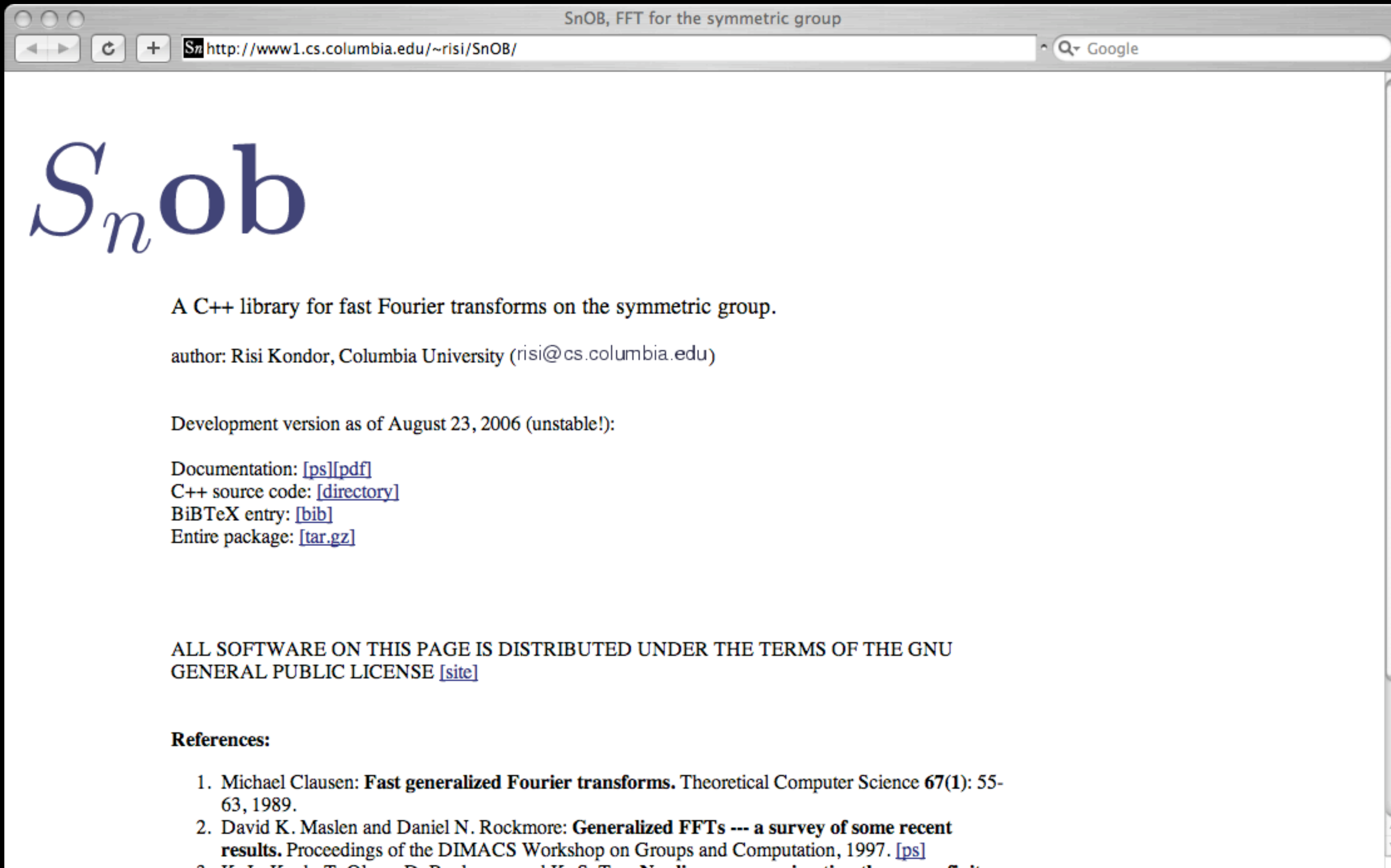
(and it's computable in $O(n^3)$ time)

$\mathbb{S}_n$





Bratelli diagram



SnOB, FFT for the symmetric group

http://www1.cs.columbia.edu/~risi/SnOB/

$S_n$ob

A C++ library for fast Fourier transforms on the symmetric group.

author: Risi Kondor, Columbia University (risi@cs.columbia.edu)

Development version as of August 23, 2006 (unstable!):

Documentation: [ps][pdf]
C++ source code: [directory]
BiBTeX entry: [bib]
Entire package: [tar.gz]

References:

1. Michael Clausen: **Fast generalized Fourier transforms.** Theoretical Computer Science **67(1)**: 55-63, 1989.
2. David K. Maslen and Daniel N. Rockmore: **Generalized FFTs --- a survey of some recent results.** Proceedings of the DIMACS Workshop on Groups and Computation, 1997. [ps]
3. K.-L. Kueh, T. Olson, D. Rockmore and K.-S. Tan: **Nonlinear approximation theory on finite**

http://www.cs.columbia.edu/~risi/SnOB

SnFourierTransform.hpp

SnFourierTransform.hpp:1 <No selected symbol>

```cpp
#include <vector>

#include "base.h"
#include "Matrix.hpp"
#include <sstream>
#include "Sn.hpp"
#include "SnFunction.hpp"
#include "StandardTableau.hpp"

using namespace std;

class Sn::FourierTransform: FiniteGroup::FourierTransform{

public:

  friend class Sn::Function;
  friend class Sn::Ftree;

  FourierTransform(const Sn& _group);
  FourierTransform(const Sn& _group, int dummy):group(&_group),n(_group.n){};
  FourierTransform(const Sn& _group, const vector<Matrix<FIELD >*> matrices);
  FourierTransform(const Function& f);
  ~FourierTransform();

  Function* iFFT() const;

  FIELD operator()(const StandardTableau& t1, const StandardTableau& t2) const;

  double norm2() const {double result; for(int i=0; i<matrix.size(); i++) result+=1; return result;}

  string str() const;

  vector<Matrix<FIELD >*> matrix;


private:

  void fft(const Sn::Function& f, const int offset);
  void ifft(Sn::Function* target, const int _offset) const;

  const int n;
  const Sn* group;

};
```

Sn::Irreducible

Represents an irreducible representation ρ_λ of S_n.

Parent class: FiniteGroup::Irreducible

CONSTRUCTORS

Irreducible(Sn* G, Partition& lambda)
: Construct the irreducible representation of the symmetric group G corresponding to the partition lambda.

MEMBER FUNCTIONS

Matrix<FIELD >* rho(const Sn::Element& sigma)
: Returns $\rho(\sigma)$, the representation matrix of permutation sigma in Young's orthogonal representation.

FIELD character(const Partition& mu)
: Returns $\chi(\mu)$, the character of this representation at permutations of cycle type μ.

void computeTableaux()
: Compute the standard tableaux of this irreducible if they have not already been computed. Because this is an expensive operation, it is postponed until some function is called (such as rho or character) which requires the tableaux of this particular irreducible. computeTableaux() is called automatically by these functions, and once the tableaux have been computed they are stored for the lifetime of the Irreducible.

StandardTableau* tableau(const int t)
: Return a new standard tableaux of index t. This works even if tableauV has not been computed.

void computeYOR()
: Compute and store the coefficients (2.5) and (2.6) in Young's orthogonal representation for all adjacent transpositions τ_k and all tableau t of shape λ. Because this is an expensive operation, these coefficients are not normally computed until they are demanded by functions such as rho or character. computeYOR() is called automatically by these functions, and once the tableaux have been computed they are stored for the lifetime of the Irreducible. computeYOR() also requires the tableaux, so it calls computeTableaux() if those have not been computed yet.

void applyCycle(const int j, Matrix<FIELD>& M[, int m])
void applyCycle(const int j, Matrix<FIELD>& M[, int m])

4. Experiments

- For n up to about 300, the skew spectrum can be computed in fractions of a second.

- For small graphs (n~5) it's complete!

- For n~100 good for learning tasks.

	MUTAG	ENZYME	NCI1	NCI109
Number of instances/classes	600/6	188/2	4110/2	4127/2
Max. number of nodes	28	126	111	111
Reduced skew spectrum	**88.61** (0.21)	25.83 (0.34)	**62.72** (0.05)	**62.62** (0.03)
Random walk kernel	71.89 (0.66)	14.97 (0.28)	51.30 (0.23)	53.11 (0.11)
Shortest path kernel	81.28 (0.45)	**27.53** (0.29)	61.66 (0.10)	62.35 (0.13)

Conclusions

- Reduced the problem of representing graphs to an abstract algebraic problem.

- Being restricted to a homogeneous space makes it easy to compute the skew spectrum but also collapses its size.

- Surprisingly, just 49 scalar invariants seem to be able enough to do the job (compressed sensing).

- Natural question: what about labeled graphs?